



Bernardo Melo Pires Soares de Albergaria

Licenciatura

Caracterização Experimental de Micro Comportamentos

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Raul Eduardo Capelo Tello Rato, Professor Auxiliar,
Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa

Co-orientador: José António Barata de Oliveira, Professor Auxiliar,
Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa

Júri

Presidente: Paulo Miguel de Araújo Borges Montezuma de Carvalho
Arguente: Nuno Filipe Silva Veríssimo Paulino



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2016

Caracterização Experimental de Micro Comportamentos

Copyright © Bernardo Melo Pires Soares de Albergaria, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Mais que tudo, aos meus pais

AGRADECIMENTOS

Um grande sentido agradecimento a todos aqueles, professores e amigos, pelo contributo que deixaram, de uma forma ou de outra, no meu crescimento ao longo destes anos.

Um agradecimento à instituição FCT-UNL pelos seus valores, os quais se manifestam em todo o ambiente que me acompanhou nestes últimos anos.

Um agradecimento ao Departamento de Eletrotecnia por todo o conhecimento transmitido.

Um agradecimento ao professor doutor Jorge da Glória, por todo o tempo que despendeu, durante o elaborar desta dissertação, a ensinar psicologia a engenheiros.

Um agradecimento também a todos aqueles que aguentaram a 1 hora e meia de experiência. Um obrigado pelos seus dados e pela sua paciência.

RESUMO

É proposta uma solução para um problema de estimação de um conjunto de sinais desconhecidos em ruído aditivo.

O ruído é considerado gaussiano branco e toda a modelação da solução apresentada depende deste facto.

O teste experimental consiste na utilização do método proposto para a estimação de sinais EEG e EMG lidos, associados a um determinado nível de agrado. Os estimadores calculados são utilizados para a classificação de intervalos temporais conhecidos dentro dos dados EEG recolhidos.

O interesse do problema está em, caso se obtenha resultados significativos, poder-se utilizar o método criado sobre dados EEG e EMG para a inferência de níveis de agrado instantâneo.

Não tendo nenhuma descrição probabilística do sinal de nível (i.e. sinal a estimar por cada nível) então a estimação de máxima verosimilhança [22] é a solução adotada para o problema da estimação.

A classificação é realizada calculando a correlação entre os intervalos temporais e os estimadores.

As experiências realizadas baseiam-se em apresentar um conjunto de imagens IAPS [29] previamente classificadas. Enquanto o slide show é apresentado é realizada a recolha dos dados do eeg e de 8 elétrodos sobre músculos faciais e da nuca. É efetuada aquisição de dados adicional para monitorizar o momento da transição dos slides.

Os resultados obtidos sobre dados de teste são concordantes com as expectativas teóricas. No entanto, a aplicação do pipeline de processamento criado sobre os dados EEG e EMG não demonstrou relação entre os resultados obtidos e os níveis das imagens IAPS.

No entanto, a análise de dados de testes realizados sobre o hardware permitiu verificar situações problemática. Um conjunto de possíveis soluções são apresentadas na conclusão deste documento.

Palavras-chave: processamento sinais eletroencefalografia eletromiografia detecção estimação agrado

ABSTRACT

This document presents a proposal of a solution to a problem of estimation of unknown signals in additive noise.

The noise is considered to be white Gaussian and the modelling of the solution depends on this fact.

The experimental test proposal is the application of the created method for the estimation of EEG and EMG signals associated with a determined pleasure level. The calculated estimates are used in the classification of known EEG time intervals.

It is an interesting problem due to the possibility of obtaining significant results, and the consequential possibility of inferring instant pleasure levels from EEG and EMG data. Since no probabilistic description of the signals to estimate is known, maximum likelihood estimation [22] is the adopted solution.

Classification of the time intervals is calculated using the correlation between the intervals and the calculated estimators.

The experimental design consist of showing previously classified IAPS [29] images. During the presentation of the slide show the EEG and 8 facial/neck electrodes data is acquired. Additional data acquisition is done with the intent of monitoring the slide transition moments.

Results obtained from test data are concordant with the theoretical expectancies. Application of the created pipeline over the EEG and EMG data didn't show any relation between the IAPS levels and the obtained results.

Analysis of hardware test data shows some troublesome situations. A set of possible solutions is presented in the conclusion of this document.

Keywords: signal processing electroencephalography electromyography detection estimation pleasure

ÍNDICE

Lista de Figuras	xvii
Lista de Tabelas	xix
Listagens	xxi
Siglas	xxiii
1 Introdução	1
1.1 Apresentação do Problema	1
1.2 Abordagem	4
1.3 Resultados Esperados	7
1.4 Contribuições	8
1.5 Capítulos Seguintes	8
2 Estado da Arte	11
2.1 Fundamentos em Psicologia	11
2.1.1 Modelos Tri-Dimensionais	11
2.2 International Affective Picture System	14
2.3 Métodos de Avaliação	15
2.3.1 Diferencial Semântico	15
2.3.2 Self-Assessment Manikin	15
2.3.3 Avaliação por Sinais Biométricos	16
2.4 Conclusões	20
3 Fundamentos Teóricos	21
3.1 Descrição formal do problema	22
3.2 Estimulo	23
3.2.1 Sequências Maximamente Aleatórias	23
3.3 Modelo	25
3.3.1 Estimação de Máxima Verosimilhança	25
3.4 Detecção	26
3.4.1 Votação	26

4	Método Experimental	29
4.1	Requisitos Pré-experimentais	29
4.2	Distribuição das Tarefas	30
4.3	Cenário Experimental	30
4.4	Procedimento Experimental	31
4.5	Realização do Processamento	35
5	Implementação	37
5.1	Slide-Show	37
5.2	Fototransistores	39
5.2.1	Sincronização Temporal	39
5.3	Equipamento e Cenário	40
5.4	Convenções de Programação	40
5.4.1	Nomenclatura	40
5.4.2	Funcionalidade	41
5.5	Organização dos Dados	42
5.6	Pipeline	43
5.6.1	Faz00 - Extrai Sinal Fototransistor	44
5.6.2	Faz01 - Load Data	46
5.6.3	Faz02 - Descobre Limites	46
5.6.4	Faz02.5 - Filtra Canais	46
5.6.5	Faz03 - Combina Canais	47
5.6.6	Faz04 - Segmenta Canais	47
5.6.7	Faz05 - Calcula Estimadores	48
5.6.8	Faz05.5 - Subtrai o Próprio	48
5.6.9	Faz06 - Aplica Estimadores	48
5.7	Elementos Variáveis	49
5.8	Software de Teste	49
5.8.1	Fazxx - Auxiliar de Teste	50
6	Análise de Resultados	53
6.1	M-sequência gerada	53
6.2	Considerações Globais	55
6.3	Validação do Software	56
6.3.1	Testes com sinais de nível gerados	56
6.3.2	Teste em Fontes de Correlação Aproximadamente Nula Expectável	59
6.4	Resultados em Participantes Experimentais	60
6.5	Discussão de Resultados	65
7	Conclusão	69
7.1	Contribuições	69
7.2	Trabalho Futuro	70

Bibliografia	71
A Convenções de Escrita	77
A.1 Nomenclatura Matemática	77
A.2 Indices	77
A.3 Outras Convenções	77
B Declaração de Consentimento Informado	79
C Pipeline de Processamento - Código Matlab®	81
C.1 Pipeline	81
C.2 Faz 00	85
C.3 Faz 01	87
C.4 Faz 02	88
C.5 Faz 02.5	90
C.6 Faz 03	93
C.7 Faz 04	94
C.8 Faz 05	95
C.9 Faz 05.5	97
C.10 Faz 06	100
C.11 Faz xx	102
D Pipeline - Fluxo de dados	105
E Filtros	107
F M-Sequência e Slide-Show	109
G Resultados Adicionais - Single Bucket	113
H Temporização das 3 Experiências	115
I Resultados Adicionais - Participantes Experimentais	117
J Resultados da Primeira Parte	121

LISTA DE FIGURAS

1.1	Lobos Cerebrais. Azul - Lobo Frontal. Verde - Lobo Temporal. Vermelho - Lobo Occipital. Amarelo - Lobo Parietal	2
1.2	Ciclo de cada simbolo - a vermelho processos informáticos e a azul processos biológicos	6
2.1	Modelo tri-dimensional de prazer, excitação e dominância [2]	14
2.2	Self Assesement Manikin [5]	16
3.1	Diagrama representativo da aplicação de um filtro adaptado sobre o sinal observado.	21
3.2	Realização do polinómio $x^6 + x + 1$ em LFSR	24
4.1	Posicionamento dos 32 elétrodo + 2 elétrodo de referencia CMS e DRL [10]	34
5.1	Slides exemplificativos de “refresh” - canto superior esquerdo -, imagem IAPS - canto superior direito -, transição - canto inferior esquerdo - e imagem não classificada - canto inferior direito.	38
5.2	Conversor analógico digital Biosemi [3] - esquerda. Touca para colocação dos elétrodo - direita	40
5.3	Cenário experimental. Mais especificamente, a cadeira onde o participante se senta e o computador onde é mostrado o slide-show	41
5.4	Diagrama representativo do método utilizado	44
5.5	Diagrama de Execução do Pipeline	45
6.1	Autocorrelação circular das 3 m-sequências geradas	53
6.2	Autocorrelação circular da sequência gerada a partir das 3 m-sequências	55
6.3	Valor médio de assertos de todos os canais. Gráficos para cada um dos dois intervalos diferentes e para a média entre os resultados de cada um.	56
6.4	Figura puramente representativa da semelhança entre os estimadores e os sinais gerados quando o SNR é alto - 40dB para o caso da figura. A vermelho e azul - os segmentos que derão origem aos estimadores. A preto - os estimadores e os sinais de nível. Cada linha de gráficos é um canal/nível diferente.	58
6.5	Segmentos e estimadores dos melhores niveis dos melhores canais - EXP01	62
6.6	Segmentos e estimadores dos melhores niveis dos melhores canais - EXP02	64

6.7	Segmentos e estimadores dos melhores níveis dos melhores canais - EXP03 .	66
B.1	Declaração de consentimento informado para uma experiência genérica - Página traseira	79
B.2	Declaração de consentimento informado para uma experiência genérica - Página frontal	80
D.1	Fluxo de dados do Pipeline. A bold ficheiros de dados criados. A texto normal ficheiros de dados utilizados. Nomes a azul são modificados e nomes a vermelho não são utilizados.	106
E.1	Característica de magnitude e fase do principal filtro utilizado.	107
E.2	Característica de magnitude e fase dos outros filtros utilizados.	108
F.1	Slide-Show criado a partir da mensagem da tabela anterior. Cada imagem corresponde a um nível de agrado proveniente da tabela.	111
H.1	Momentos detetados para início dos slides para cada experiência - apenas relativo à parte da experiência em que o participante vê os slides. Também é mostrado o sinal kAzul.	116

LISTA DE TABELAS

2.1	Autoria das nomenclaturas.	12
2.2	Relação entre as dimensões de Mehrabian e Russel e três nomenclaturas diferentes [2].	12
2.3	Operacionalização das três dimensões.	13
6.1	Picos e mínimos na autocorrelação circular da sequência 1	54
6.2	Picos e mínimos na autocorrelação circular da sequência 2	54
6.3	Picos e mínimos na autocorrelação circular da sequência 3	54
6.4	Picos e mínimos na autocorrelação circular da sequência final	55
6.5	Número de pontos mínimos representativos da recta de valor esperado dos dois intervalos 6.3	57
6.6	Correlacao entre os estimadores e os sinais de nível gerados para os canais [1 2 3 4 5 6 7 8]	59
6.7	Resultados para o Single Bucket	59
6.8	Resultados para testes sobre ruído branco	60
6.9	Resultados para a EXP01	61
6.10	Probabilidade de Erro e Acertos Para cada Nível - EXP01	61
6.11	Resultados para a EXP02	63
6.12	Probabilidade de Erro e Acertos Para cada Nível - EXP02	63
6.13	Resultados para a EXP03	63
6.14	Probabilidade de Erro e Acertos Para cada Nível - EXP03	65
F.1	Conjunto de 3 sequências geradoras da mensagem de 3 bits e respetiva mensagem criada.	111
G.1	Resultados adicionais para o Single Bucket	113
I.1	Resultados adicionais para a experiência EXP01	117
I.2	Resultados adicionais para a EXP02	118
I.3	Resultados adicionais para a EXP03	119
J.1	Resultados para a primeira parte da experiência EXP01 - participante não vê os slides	121

J.2	Resultados para a primeira parte da experiência EXP02 - participante não vê os slides	121
J.3	Resultados para a primeira parte da experiência EXP03 - participante não vê os slides	121

LISTAGENS

C.1 Faz 00	85
C.2 Faz 01	87
C.3 Faz 02	88
C.4 Faz 02.5	90
C.5 Faz 03	93
C.6 Faz 04	94
C.7 Faz 05	95
C.8 Faz 05.5	97
C.9 Faz 06	100
C.10 Faz xx	103

SIGLAS

BCI Brain-Computer Interface.

BDI Body-Device Interface.

BoCI Body-Computer Interface.

EEG Eletroencefalograma.

EMG Eletromiograma.

ERP Event Related Potencial - Potencial de Eventos Relacionados.

IAPS Internacional Affective Picture System.

SAM Self-Assessement Maniki.

SNR Signal to Noise Ratio - Rácio Sinal Ruído.

INTRODUÇÃO

“Problems cannot be solved by the level of awareness that created them.”, Albert Einstein

1.1 Apresentação do Problema

Ao longo dos tempos tem-se verificado que com a invenção de novos instrumentos de medida existe a possibilidade de observar fenómenos que sempre estiveram presentes. Um exemplo deste facto é a invenção da eletroencefalografia.

As primeiras gravações, por Hans Berger (1873 - 1941) [47], usando um eletroencefalograma ([EEG](#) - electroencephalography) remontam a 1924. No entanto, os primeiros relatórios remontam a 1929. A sua descoberta permitiu a leitura e gravação não invasiva de atividade elétrica ao longo do escalpe.

A técnica do [EEG](#) consiste na captura e registo de variações de tensão provocadas pelas correntes iónicas nos neurónios [37]. Grande parte da investigação realizada desde a sua invenção foca-se no estudo da epilepsia, em parte devido à influência evidente dos ataques sobre as leituras realizadas [32].

A invenção do EEG permitiu a observação de atividade elétrica cerebral diretamente relacionada com estímulos percebidos ou algum tipo de ação motora a que se chama de potenciais de eventos relacionados [31] ([ERP](#) - Event-related potential). Regra geral são caracterizados pela polaridade e a sua ocorrência no tempo. Por exemplo, o [ERP](#) P300 é assim chamado por ser um pico positivo que ocorre aproximadamente 300 milissegundos após a interação com um estímulo. Os primeiros estudos sobre o potencial elétrico P300 remontam aos anos 60 [9]. Associa-se a manifestação do potencial a processos de tomada de decisões. Este fenómeno é mais facilmente medido no lobo parietal [27], o qual pode

ser distinguido com auxílio da imagem 1.1 na qual se pode observar uma versão simplificada do mapa cerebral (apenas se distingue os diferentes lobos).

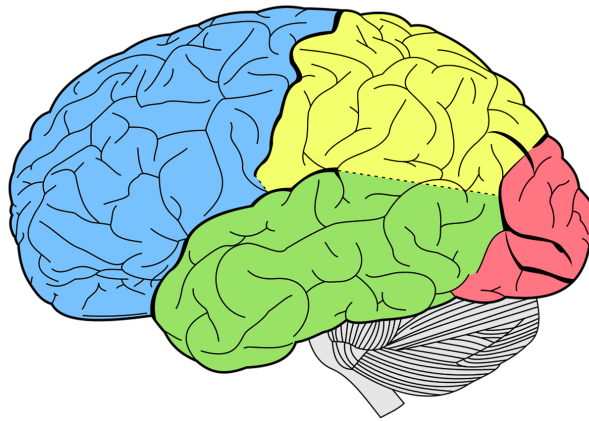


Figura 1.1: Lobos Cerebrais. Azul - Lobo Frontal. Verde - Lobo Temporal. Vermelho - Lobo Occipital. Amarelo - Lobo Parietal

Com a breve cronologia de avanço científico apresentada é possível ter uma noção de como novos dados levam à inferência de novo conhecimento.

A possibilidade de observar ERP's como o P300 permitiu uma nova vaga de estudos, em que a informação emocional é inferida através da interpretação de dados provenientes de um EEG.

Em 1996, Alessio Mini et al. [35] verificou que existe influência de estímulos emocionais (i.e. agradáveis e desagradáveis) na amplitude do potencial P300. Outros estudos anteriores chegaram a conclusões semelhantes [27, 41].

É possível verificar, desde estudos anteriores, como o conteúdo emocional se manifesta nos sinais EEG lidos - investigação relativa a esta relação será abordada com mais detalhe no capítulo 2. Portanto, dadas as conclusões extraídas dos artigos referidos, torna-se pertinente perguntar: existirá a possibilidade de criar um instrumento de medida para valores emocionais ?

Para tal são necessários melhores sistemas de aquisição de dados, mas também bons sistemas de inferência de dados. É ainda necessário saber como extrair informação consistente e fiável dos dados recolhidos.

Agora é um bom momento para fazer uma breve distinção entre dados e informação [1]:

- Dados - São conteúdos não processados. Dados são factos que podem gerar informação. No caso mais comum que se irá falar, um dado será, por exemplo, um conjunto de leituras EEG ao longo de um intervalo de tempo. Estes dados pressupõem processamento futuro.

- Informação - São dados processados. É a atribuição de significado aos dados processados. É informação que pode ser utilizada e que seja significativa de alguma forma.

De uma forma mais concreta, nesta dissertação pretende-se criar um sistema que consiga recolher dados provenientes do corpo humano e processá-los num sistema com poder de computação de modo a extrair um valor de agrado. E isto pressupõe uma abordagem ao que se irá chamar de Interface Corpo-Dispositivo (Body-Device Interface - [BDI](#)), mas, devido às limitações da tecnologia disponível, os dados serão processados num computador, de forma a se ter resultados em tempo útil. Portanto, na prática, será uma Interface Corpo-Computador (Body-Computer Interface - [BoCI](#)). A ideia é simplesmente não cair em abordagens limitativas. Esta abordagem ao contrário da comumente utilizada - Interface Cérebro-Computador (Brain-Computer Interface - [BCI](#)) [[15](#)] - permite que se considere a utilização de dados provenientes de outras áreas do corpo para a criação de informação.

O software criado terá de conseguir inferir e identificar micro-comportamentos. Por conveniência, comportamento será definido como a resposta a um estímulo. Por micro-comportamento entenda-se que se está a falar a nível dos sinais lidos e não do comportamento psicológico (em escala macro) da pessoa.

Mais especificamente, o objetivo é, através de um conjunto de elétrodos posicionados pelo escalpe e face do participante, fazer inferências, ao longo de intervalos de tempo, sobre o valor numa escala de uma dimensão: agrado. A dimensão de agrado será explicada mais ao pormenor no capítulo [2](#), no entanto é relevante referir que esta dimensão foi escolhida de entre três (agrado, excitação e dominância). Destas três dimensões o agrado apresentava um desvio da média menor para maior parte das imagens. Estas dimensões são quantificadas através de métodos da área da psicologia, como o Self-Assessment Manikin ([SAM](#)) [[5](#)] e o Diferencial Semântico [[21](#)], ambos explicados em maior pormenor no capítulo [2](#).

Resumidamente, estes métodos ou são baseados na exposição de imagens que representem diferentes pontos da dimensão a inferir (e.g. Para a dimensão do agrado são mostrados 5 bonecos, o primeiro com uma expressão facial feliz - para agrado máximo. O sorriso vai desaparecendo ao longo das imagens, até que o boneco assume uma expressão de tristeza - para desagrado máximo) ou na utilização de outras palavras e os seus opostos (e.g. quente e frio, branco e preto), associados à dimensão a medir.

Estes métodos permitiram a classificação do mais diverso material. Mais concretamente, o [SAM](#) permitiu a classificação de um conjunto grande de imagens, utilizando o input de um numero grande de participantes e disponibilizadas online- International Affective Picture System ([IAPS](#)) [[29](#)]. Também no capítulo [2](#), estes métodos serão discutidos em maior detalhe.

As imagens serão o conjunto de estimação a utilizar- imagens [IAPS](#).

Esta informação - estímulo e significado (i.e. nível de agrado) - poderá ser utilizada para criar o modelo para a tomada de decisões (sobre um numero de escolhas possíveis).

A abordagem clássica a este tipo de problemas, é tratá-los como problemas de classificação. Basicamente são utilizados dados que já venham classificados ao longo de uma dimensão (e.g. agrado, excitação) e o software tenta encontrar padrões ao longo deles. São utilizados algoritmos de classificação em que o investigador pode controlar alguns parâmetros do seu funcionamento, tendo alguma margem de manobra dentro do método escolhido.

Estas abordagens são capazes de atingir resultados satisfatórios (acima de 70% [30] de taxa de sucesso), apesar de realizados em poucos participantes, por vezes são realizados apenas sobre o próprio investigador.

Existe no entanto um problema crucial em processamento de dados EEG. Apesar de se obter resultados satisfatórios, é difícil reproduzir os resultados. Ou seja, o mesmo método não atinge as mesmas taxas de sucesso quando aplicado a diferentes participantes ou em diferentes condições.

Existe então a necessidade de uma metodologia que seja segura e fiável. Isto pode garantir-se, ou através de um método que obtenha taxas de decisões corretas semelhantes em todos os testes, ou através de um método que esteja pronto a lidar com situações em que não se pode garantir essas mesmas taxas de sucesso.

1.2 Abordagem

É de toda a conveniência começar pela explicação direta de como resolver este problema, de forma a que se evitem futuras confusões:

Será criado um slide-show com diferentes imagens da base de dados IAPS e imagens de marcas - não classificadas. Apenas as imagens IAPS constituem a mensagem a transmitir. As imagens IAPS terão níveis de agrado associados - de 1 (muito desagradável) a 9 (muito agradável), exceto o nível neutro 5. O nível neutro não é considerado por comodidade - assim garante-se um numero par de níveis (algo útil, uma vez que os métodos de criação das imagens são binários), garantindo também a existência do mesmo numero de níveis agradáveis e desagradáveis.

Por questões de simplicidade na explicação do método, convém considerar os elementos da mensagem como os níveis de agrado e não as imagens em si.

1. O slide-show irá conter a mensagem e será mostrado ao participante experimental;
2. Os sinais da superfície do escalpe e da face são recolhidos;
3. Estes sinais irão ser utilizados para criar um sistema de decisão entre os 8 níveis de agrado, para cada intervalo de exposição a um símbolo da mensagem (i.e. imagem IAPS);

4. Após verificar que a mensagem detetada é igual à transmitida (ou significativamente semelhante), pode-se utilizar o mesmo sistema de decisão para classificar as imagens de marcas.

Também por questões de simplicidade, o participante, enquanto sistema, é considerado como sendo invariável no tempo durante o período de exposição do slide-show. Por isso as imagens de marcas são mostradas intercaladas com as imagens IAPS - de forma a que o mesmo sistema de decisão possa ser utilizado.

Independentemente da abordagem, há que ter consciência de duas possibilidades diferentes:

- H_0 : Os estímulos não se manifestam nos sinais biométricos.
- H_1 : Os estímulos manifestam-se em alguma componente (e.g. intervalo de tempo/frequência) de algum sinal biométrico recolhido. Será então possível detetar e classificar o estímulo através dos dados recolhidos à superfície do corpo;

O software criado terá que ser capaz de distinguir o participante experimental como válido ou não para a experiência. A abordagem é típica em tele-comunicações:

- O participante é considerado como um canal de comunicação;
- Uma mensagem é transmitida através do canal;
- Se se conseguir recuperar no recetor uma percentagem significativa da mensagem, então o canal está aberto. Caso contrário, o canal está fechado e não pode ser utilizado na classificação de valores desconhecidos;
- O software só pode ser utilizado como ferramenta se o sujeito puder ser utilizado como canal.

Esta será a abordagem relativamente à avaliação do participante.

Quanto à receção da mensagem em si, deverá ser considerada como sendo um problema de deteção. Num problema clássico de deteção:

1. Existe um sinal que é transmitido - $x[n]$;
2. Ao longo do percurso, este sinal é corrompido por ruído - $x[n] + \eta[n]$;
3. O recetor tem que reagir de alguma forma caso receba o sinal em questão.

No problema de deteção abordado:

1. Existe um conjunto de sinais diferentes que são transmitidos - níveis IAPS - $N[n] \in \{1, 2, 3, 4, 6, 7, 8, 9\}$ os quais estarão relacionados com um conjunto de imagens IAPS;

2. Estes sinais sofrem de transdução ao entrarem no canal (i.e. sujeito experimental), e é esperado que o sinal transduzido gere uma manifestação do nível de agrado - $s(t)$;
3. Esta manifestação, caso exista, surge num canal que a corrompe com ruído - $s(t) + \eta(t)$.
4. Os sinais são capturados, mas os níveis (i.e. símbolos da mensagem) não são detetados enquanto não forem processados - $N'[n]$.

Convém diferenciar tempo discreto - n - de tempo contínuo - t - nos processos informáticos e nos processos biológicos, respetivamente.

A hipótese não nula - $H1$ - implicará que exista um sinal $s(t)$ com uma elevada correlação com os níveis transmitidos $N[n]$. A relação entre $N[n]$ e $M[n]$ foi assegurada pela investigação realizada pelo [IAPS](#), a qual será referida com mais detalhe no capítulo 2.

Caso exista relação entre $s(t)$ e $N[n]$, então, em teoria, caso se capte $s(t)$, de alguma forma será possível recuperar a mensagem original - $N[n] = N'[n], \forall n$.

Pode-se reformular a hipótese anterior como:

- $H0$: Não existe sinal $s(t)$ que esteja diretamente associado a $N[n]$;
- $H1$: A existência de um sinal $s(t)$ permite a recuperação da mensagem original $N[n]$.

A abordagem ao problema pode ser classificada como um problema de deteção de classe II, segundo Hippenstiel [22] - Sinais com parâmetros desconhecidos em ruído aditivo.

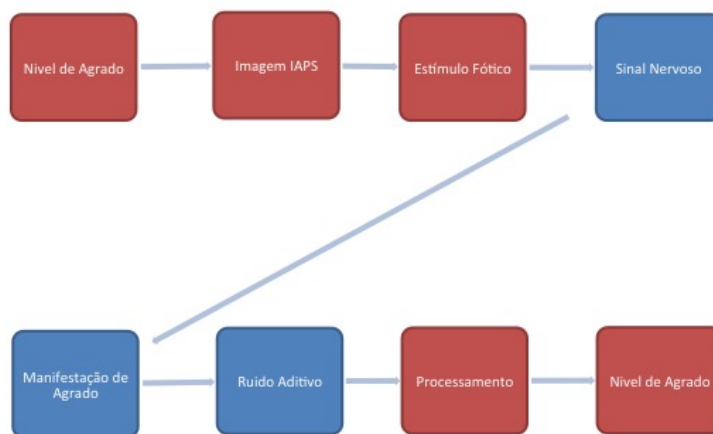


Figura 1.2: Ciclo de cada símbolo - a vermelho processos informáticos e a azul processos biológicos

A figura 1.2 permite observar todo o percurso pelo qual um símbolo (i.e. nível [IAPS](#)) passa desde que é transmitido até que é detetado e classificado. A imagem também

distingue processos biológicos (a azul) de processos informáticos, de forma a que se entenda o que é do âmbito da neuro ciência e o que é do âmbito de processamento de sinais.

O ruído é informação que não é relevante para os valores que se querem estimar - agrado. Neste caso será toda a informação presente nos sinais lidos que não é uma manifestação do agrado que o participante sente ao visualizar cada imagem. Portanto o problema vai além do clássico problema de deteção. O recetor não capta um sinal conhecido, uma vez que este sofre transdução ao entrar no canal humano. Este sinal desconhecido é corrompido com ruído.

De forma a lidar com os sinais desconhecido, é necessário criar estimadores para cada nível.

Um estimador pode ser visto como o sinal esperado para cada nível. Como tal, cada estimador de nível será criado através da média de todos os intervalos (da leitura EEG) referentes a esse nível.

Utilizar a média traz a vantagem adicional de aumentar a relação sinal/ruído, pelo que se está também a lidar com o problema do ruído, além do problema da estimação.

Haverá duas variáveis possíveis de manipular em cada processamento realizado: o formato dos intervalos e a banda de frequências a considerar. O formato do intervalo refere-se a quantas amostras se devem considerar antes e depois de cada imagem ter começado a ser apresentada.

1.3 Resultados Esperados

De inicio é difícil saber se o modelo proposto é o ideal para o tratamento dos dados. No entanto, grande parte do trabalho incidirá sobre a descoberta do melhor intervalo temporal e de frequências. A utilização de um número considerável de elétrodos permite a gravação da atividade de uma área maior do escalpe. Isto permitirá a distinção da eficácia de diferentes elétrodos. Por outras palavras, isto permitirá verificar que áreas do cérebro são responsáveis por sinais que estejam fortemente correlacionados com os sinais IAPS transmitidos.

Pode-se esperar um número de resultados mais fortemente relacionados com a mensagem enviada em dados provenientes de elétrodos localizados sobre o lobo temporal. Este lobo é responsável por processar informação proveniente de estímulos, extraindo significado e fazendo associações dos mesmos com memórias e emoções [44]. A utilização do P300 também passa pelo seu uso na classificação de estímulos emocionais, uma vez que se verifica uma forte relação entre a forma de onda do ERP e o conteúdo emocional apresentado [35].

Apesar de não se estar à procura da forma de onda P300, também se pode esperar melhores classificações no lobo parietal, onde este ERP se manifesta mais proeminentemente.

1.4 Contribuições

As possíveis contribuições desta dissertação serão confirmadas no final. No entanto, existe um conjunto de possíveis contributos:

- Um sistema de processamento de dados para o modelo apresentado - caso o modelo seja sustentado teoricamente e funcione como esperado para situações de teste, então pode ser utilizado para situações em que o modelo considerado se aplique;
- Um método e procedimento experimental funcional para testes sobre participantes utilizando eletroencefalografia;
- Uma metodologia que mistura conhecimento sobre a área da psicologia, engenharia e neuro ciência para tentar resolver o problema em questão;
- Em ultimo caso, um método para classificação de níveis de agrado.

1.5 Capítulos Seguintes

- Capítulo 2 - É explicado o estado da arte no que toca a sistemas de classificação de informação biométrica. São apresentados também os métodos de classificação utilizados em psicologia, assim como as noções da dimensão de agrado, dentro do contexto.
- Capítulo 3 - É feita a descrição formal do problema. Este capítulo está encarregue de dar significado matemático a todo o método que foi descrito. Está também encarregue de explicar algumas consideração que justifiquem a abordagem utilizada. São feitas as demonstrações matemáticas necessárias.
- Capítulo 4 - Neste capítulo é feita a descrição, passo a passo, de como se deverá realizar a experiência. Basicamente, este capítulo é o manual de instruções para recriar o método utilizado, desde necessidades pré-experimentais, realização da experiência e toda a interferência humana necessária para realizar o processamento sobre os dados.
- Capítulo 5 - Explica a implementação realizada. É uma explicação detalhada de como se implementou o método descrito na prática. É feita uma descrição de como se criou o slide-show. É explicado também como os dados foram organizados e como o formato dos intervalos e a banda de frequências (elementos variáveis) foram manipulados.
- Capítulo 6 - Os resultados obtidos são apresentados e analisados. É apresentado um histórico de resultados obtidos, desde os primeiros testes executados. Estes resultados serão analisados quanto à sua validade e quanto às conclusões que se podem retirar dos mesmos.

- Capítulo 7 - Considerações finais sobre o trabalho realizado. É feita uma avaliação do método utilizado à luz dos resultados obtidos. É o capítulo em que se admite ou exclui a hipótese experimental apresentada.

ESTADO DA ARTE

2.1 Fundamentos em Psicologia

2.1.1 Modelos Tri-Dimensionais

“Human behavior is predictable, but, as in physical science, accurate prediction hinges on the correctness of underlying theoretical assumptions.”, Douglas McGregor [33]

Um dos principais problemas a ser descrito aquando da descrição da investigação feita é a falta de rigor ao descrever situações emocionais. Obviamente, quando rotuladas de tal forma, os dados poderão ser concordantes ou não com as nomenclaturas utilizadas. A dificuldade está em estabelecer comparações entre a investigação feita e o nosso caso específico. Consequentemente, é necessário falar sobre as três principais dimensões e as alternativas sugeridas para as mesmas.

Para clarificar as ideias quanto a este tópico ir-se-á recorrer ao trabalho realizado por Bakker et al [2], que, após anos de investigação nesta área e propostas de novos modelos, tenta resumir e clarificar as ideias base. Revisita, para tal, o modelo sugeridos em 1974 em “An approach to environmental psychology” por Mehrabian e Russel [34] - Prazer, Excitação e Dominância - e o modelo sugerido por Osgood et al [21] - Avaliação, Atividade e Potência. Faz também a analogia das três dimensões com o Modelo de Atitudes (ABC) : Affect, Behaviour and Cognition e com a tríade Sentimento, Pensamento e Ação que configura as três funções da alma, segundo Platão. Apenas pela noção prévia do significado das palavras destas triplas, é possível começar a imaginar algumas relações óbvias.

Na tabela 2.2 pode-se ver a relação entre as 4 nomenclaturas sugeridas e na tabela 2.1 é atribuída a autoria às mesmas.

Dimensões	Mehrabian e Russel [34]
Fatores	Osgood et al [21]
ABC	
Funções da alma	Platão

Tabela 2.1: Autoria das nomenclaturas.

Dimensões	Fatores	ABC	Funções da alma
Prazer	Avaliação	Afetivo	Sentimento
Excitação	Atividade	Cognitivo	Pensamento
Dominância	Potência	Conotativo	Ação

Tabela 2.2: Relação entre as dimensões de Mehrabian e Russel e três nomenclaturas diferentes [2].

O artigo tenta também explicar a importância de incluir a terceira dimensão - Dominância - nos modelos considerados.

Os modelos surgem da necessidade de descrever uma grande gama de emoções diferentes. Haverá algum conjunto de dimensões mais reduzido que consiga representar e que seja responsável pelo que se entende como sendo diferentes emoções? Esta necessidade surge principalmente em psicologia ambiental. A psicologia ambiental pretende estudar o comportamento do ser humano no ambiente que o rodeia. Pode dar-se o exemplo de estudos pioneiros sobre tópicos como a relação entre Homem, o Sol e a Lua por Willy HellPach.

De forma a classificar corretamente as emoções e a permitir que se compreenda as nomenclaturas utilizadas é necessário alguma clarificação dos conceitos. De acordo com a interpretação proposta por Mehrabian e Russel em 1974 [34], prazer, excitação e dominância são três dimensões (contínuas) não relacionadas. Operacionalizaram ¹ estas três dimensões utilizando adjetivo e conceitos como os apresentados na tabela 2.3 ². Mehrabian e Russel estabelecem a ligação com os termos anteriormente propostos anteriormente por Osgood et al [21] - avaliação, atividade e potência.

- Avaliação - Prazer: Ambas se referem a considerações emocionais relativas ao ambiente. Avaliação é descrita por adjetivos e conceitos como bom-mau, otimismo-pessimismo, positivo-negativo. Prazer refere-se, em geral, a sentimentos positivos ou negativos;
- Atividade - Excitação: Excitação é descrita como sendo um indicador de atividade mental como se pode depreender dos adjetivos referidos na tabela 2.3. Atividade

¹Operacionalização é o método utilizado quando são necessárias medidas de um fenómeno que não pode ser diretamente medido. Refere-se a uma fuzzificação de um conceito, no sentido de se estabelecer uma gradação de um conceito em outros "sub-conceitos".

²Tradução para o mesmo nome (satisfeito) nos dois casos (dimensão de prazer). Traduzido de: pleased-annoyed, satisfied-unsatisfied.

Prazer	feliz-infeliz satisfeito-aborrecido satisfeito-insatisfeito
Excitação	excitação frenética-sono estimulado-relaxado excitado-calmo acordado-sonolento
Dominância	dominância-submissão controle, influência e autonomia

Tabela 2.3: Operacionalização das três dimensões.

é operacionalizada utilizando termos como ativo-passivo, excitado-calmo, quente-frio, podendo ser definida como uma “medida” de atenção. Está também associada a rispidez física, não sendo apenas um conceito psicológico. Excitação está associada ao aumento de atenção;

- Potência - Dominância: Potência é descrita utilizando conceitos como leve-pesado, feminino-masculino, suave-severo. Dominância está relacionada com a necessidade de controle, normalmente determinada por barreiras sociais ou físicas. No entanto, é possível reparar que os adjetivos com que se descrevem Potência e Dominância são bastante diferentes no seu significado, pelo que se pode inferir que poderão ser duas dimensões não relacionadas, como é comprovado por resultados experimentais.

Os conceitos prazer e excitação também são associados respetivamente a conceitos como ordem e variação³. A relação vem quando se considera um outro fator. A harmonia poderá ser interpretada como uma consequência de estados baixos/intermédios da dupla prazer-excitação, enquanto que a desarmonia estará relacionada com estados extremos destas dimensões. Da mesma forma, demasiada ordem/variação ou a falta das mesmas irá levar a situações desarmoniosas.

Prazer e excitação estão também relacionados com expectativas, sendo, para a mesma situação, dependentes de sujeito para sujeito. Desvios das expectativas levam a um estado maior de excitação. Caso este desvio seja experienciado positivamente, irá ser despoletado um estado de prazer e, para o caso contrário, um estado de desprazer.

Em convívio com o ambiente em redor estamos constantemente sujeitos a estímulos. Estes estímulos são interpretados de certa forma e acabam por produzir uma resposta emocional. Evidentemente, cada sujeito diferente terá diferentes necessidades de estímulos, pelo que o prazer, a excitação e a dominância experienciadas em diferentes situações dependem das necessidades de cada sujeito. Isto dá azo a situações consideradas cómodas

³Ordem é referente à dupla caos vs. rigidez. Variação refere-se a diferentes representações dos mesmos conceitos, no que toca à sua representação particular (i.e. diferentes aparências do mesmo conceito devido ao contexto).

ou incomodas, dependendo da congruência do ambiente com as variáveis de personalidade [17].

No que toca a resultados práticos, prazer e excitação demonstram uma variação muito superior à dominância, mas, no entanto, avaliação é responsável pelo dobro da variância do que atividade ou potência. Esta discrepância entre potência e dominância, relativamente aos outros dois elementos de cada modelo poderá ser justificada devido ao uso de adjetivos não significativamente relacionados e/ou simplesmente pelo facto de serem duas dimensões não relacionadas.

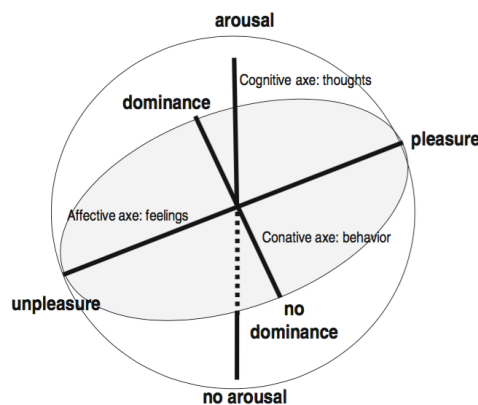


Figura 2.1: Modelo tri-dimensional de prazer, excitação e dominância [2]

Um erro clássico, visto recorrentemente, é incluir prazer ou excitação num conjunto de emoções, juntamente com conceitos mais comumente utilizados para descrição emocional - alegria, tristeza, raiva, repulsa, surpresa e medo [16].

Por exemplo, pode-se descrever raiva como sendo uma situação desagradável e de elevada excitação.

Para a investigação realizada foram escolhidas as dimensões de Mehrabian e Russel [34] devido à sua simplicidade de compreensão e devido a serem as três dimensões consideradas na base de dados de imagens IAPS. Apesar da ideia da descrição de emoções utilizando um número reduzido de dimensões (cuja conjugação das mesmas daria uma gama de emoções diferentes) surgir com Osgood, as dimensões de Mehrabian e Russel acabam por ser mais comumente utilizadas na literatura moderna.

2.2 International Affective Picture System

O objetivo do IAPS passa por fornecer uma base de dados de fácil acesso que permita uma fonte para investigação o mais fidedigna possível. Esta base de dados contém um número diverso de imagens, de situações diferentes, escolhidas de forma a que ocupem uma gama grande de valores das três dimensões - prazer, excitação, dominância.

De forma a determinar o valor tri-dimensional associado a cada imagem recorreu-se ao Self-Assessment Manikin. Foi pedido aos sujeitos, após contacto com cada imagem, que

indicassem o seu estado, escolhendo uma das imagens para cada fila do SAM ou qualquer dos espaços entre elas. Isto dá um total de nove valores (5 figuras + 4 espaços) possíveis para cada dimensão.

A base de dados utilizada consiste num conjunto de 1196 imagens divididas entre 20 sets. As imagens são de fácil compreensão do seu conteúdo, com as condições adequadas de cor e luminância.

A maior parte dos participantes são alunos de faculdade, se bem que também foram mostradas imagens a crianças de um conjunto de sets específico, às quais era dado mais tempo para avaliarem o que sentiam. As imagens foram apresentadas num slide-show em que os colaboradores tinham 6 segundos para ver cada imagem e 15 segundos para descreverem como se sentiram, atribuindo um valor relativo às filas do SAM. Um conjunto de três figuras - duas situações extremas e uma neutra - era mostrado antes de cada set. Foi utilizado também um slide de preparação entre imagens.

Os resultados podem ser observados em [29].

2.3 Métodos de Avaliação

2.3.1 Diferencial Semântico

A abordagem surge em 1958 com Osgood [21]. Pretendia-se uma forma de medir o significado de estados, acontecimentos ou conceitos. Para tal recorre-se a dois adjetivos opostos e pede-se ao sujeito que defina o seu estado através de um ponto entre os dois extremos. Isto ocorre após a operacionalização, ou seja, a escolha dos adjetivos corretos para descrever o caso em específico.

2.3.2 Self-Assessment Manikin

O Self-Assessment-Manikin [5] é um método de medir o estado do sujeito experimental utilizando um conjunto de imagens sugestivas apresentadas na figura 2.2, para cada uma das três dimensões. O método revelou resultados semelhantes ao diferencial semântico, exceto para a dominância⁴, se bem que esta mantém valores de variância relativamente baixos. O método revela-se mais poderoso que o diferencial semântico pela sua simplicidade. Enquanto que, utilizando o diferencial semântico, diferentes julgamentos têm que ser feitos (de forma a mapear o estado prazer-excitação para cada dimensão, uma vez que diferentes adjetivos são utilizados para descrever cada dimensão), o SAM só necessita de três julgamentos (um por dimensão). A abordagem é simples: Para a dimensão de prazer existe um conjunto de 5 figuras, em que uma figura semelhante a uma forma humana altera entre uma expressão facial de felicidade até uma de tristeza. Na segunda fila - excitação - a mesma figura passa de um estado agitado e explosivo (frenético) até

⁴No artigo é levantada uma questão pertinente. Uma das principais razões para que dominância seja uma dimensão tão difícil de determinar deriva do facto de muitas vezes ser complicado perceber se se está a avaliar a dominância do sujeito em relação ao estímulo ou do estímulo em relação ao sujeito.

um estado calmo. Na terceira fila, relativa à dominância, é possível observar a passagem de um estado em que a figura se encontra pequena e no meio do quadrado que a envolve, até uma figura que ocupa mais espaço que o quadrado que a envolvia.

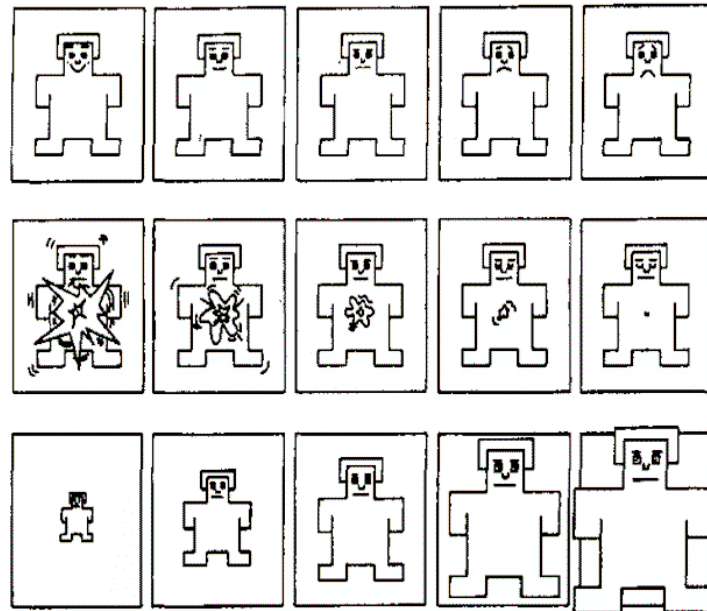


Figura 2.2: Self Assessment Manikin [5]

2.3.3 Avaliação por Sinais Biométricos

2.3.3.1 Métodos de aquisição de dados

2.3.3.2 EEG

Em 2006 a área começava a cativar peritos em computação. É de referir o trabalho de Chanel, Kronegg, Grandjean et al. [7]. Não foi encontrado registo de trabalho anterior tão semelhante com o que se está a realizar. A diferença está na dimensão que é considerada, uma vez que o método pretende avaliar excitação e não prazer (como é o caso desta dissertação). O trabalho utiliza dois classificadores para inferir entre dois ou três graus diferentes de excitação.

Os trabalhos iniciais demonstram a possibilidade de, através de processamento computacional, distinguir características emocionais.

Um trabalho interessante utiliza musica [30] para tentar amplificar os efeitos emocionais nos sinais biométricos, tornando-os mais facilmente distinguíveis. O método pretende distinguir entre quatro emoções diferentes: estado elevado de prazer, raiva, tristeza e alegria. É utilizado um algoritmo de classificação baseado numa rede neuronal. São utilizadas funções de ativação maioritariamente não lineares, sendo o objetivo final fazer o mapeamento de vetores de características nas quatro emoções referidas. Foram realizados testes em cinco participantes, obtendo-se uma percentagem de resultados corretos de

aproximadamente 70%;

De Kruif, Bas, Schaefer et. al. [12] desenvolveram um método para distinguir informação tonal e rítmica através dos dados obtidos através de um EEG. A informação extraída do EEG foi gravada durante dois segundos. Foram utilizados classificadores que utilizam informação de fase e energia do sinal. Apenas quatro participantes se submeteram aos testes, tendo sido obtida uma taxa de sucesso de 60% a 70% quando utilizadas combinações de classificadores.

Chanel [8] volta a investir na área em 2007. Desta vez, pretendendo distinguir áreas no espaço prazer-excitação. As três áreas que o método propõe distinguir são excitação positiva e negativa e neutralidade calma. Esta distinção é tornada possível recorrendo a reconhecimento de padrões através de dois classificadores. Os classificadores foram testados com base em diferentes conjuntos de características, provenientes tanto de elétrodos periféricos como do EEG em si.

A grande particularidade está na forma de estimular o colaborador. É pedido que este se recorde de situações específicas que configurem qualquer uma das três combinações de valores do espaço prazer-excitação. Os resultados revelam que mais investigação neste assunto pode vir a ser interessante e relevante.

Foi desenvolvido um método [23] através da utilização de algoritmos de aprendizagem para determinar como a atividade cerebral se relaciona com a emoção associada. Foi elaborado um conjunto de cinco emoções caracterizada num espaço bi-dimensional (prazer, excitação). Foram mostradas imagens aos participantes, distinguidas entre as cinco emoções diferentes. Os resultados são fracos quando se pretende distinguir as cinco emoções. Para a dimensão do prazer o sinal foi classificado corretamente em 32% e para a da excitação foi de 37%. No entanto quando apenas se pretendia distinguir valores extremos do espaço, 71% do prazer foi corretamente distinguido, sendo o resultado para a excitação de 81%.

Em 2009 começam a surgir diferentes ideias de aplicação para os sistemas de reconhecimento emocional. Schaaff e Schultz [43] propuseram que métodos de reconhecimento de emoções poderiam ser usados na interação humanoide-humano para que este se possa adaptar e reagir consoante a informação recebida. Para este efeito foram utilizadas 90 imagens do IAPS. As imagens são divididas em três categorias - agradáveis, neutras e desagradáveis (referindo-se à dimensão de prazer). No artigo é sugerido um método de distinção das três categorias. São utilizadas máquinas de vetores de suporte que permitem a aprendizagem supervisionada de padrões. Os resultados revelaram uma taxa média de decisões corretas de 47.11%.

A abordagem de Xu e Plataniotis [49] pretende avaliar estados emocionais, definidos por três valores - positivamente excitado, neutro e negativamente excitado. No artigo é reservada uma parte para a descrição das principais etapas da avaliação de emoções, começando pela aquisição dos dados, passando pela extração de características, até à classificação do conjunto obtido. Foram utilizadas imagens da base de dados IAPS e realizados testes em 5 sujeitos experimentais.

Neste caso específico foi utilizado o algoritmo de classificação k-Nearest Neighbour. Este método baseia-se em utilizar os vizinhos mais próximos, obtidos através de uma determinada métrica de distância (e.g. distância euclidiana). Cada vizinho “lança um voto” sobre a classificação do objeto a classificar. Neste caso específico foram considerados os votos dos cinco vizinhos mais próximos. O algoritmo obteve uma taxa média de classificação correta de 90.77%.

Yohanes, Ser e Huang [50] propõe a utilização dos coeficientes da Transformada Wavelet Discreta como características para a classificação. Estes coeficientes representam a correlação do sinal com a função de Wavelet e o sinal a analisar em diferentes instantes de tempo. Foram utilizadas imagens do IAPS e dois elétrodos para a aquisição dos dados. Para a classificação utilizou-se máquina de vetores de suporte e máquina de aprendizagem extrema. Os resultados demonstram taxas de classificações corretas superiores quando a utilização (em investigações anteriores pelo autor) dos coeficientes de Fourier. Isto comprova a importância de não abdicar da informação temporal para a classificação de sinais de um EEG.

A investigação realizada por Islam et al [24] tem por objetivo classificar o estado do cérebro de entre 7 estados possíveis - relaxado, pensamento neutro, nostálgico (a relembrar algo), em movimento, prazer, medo e a ouvir música. A análise utiliza as medidas estatísticas de curvatura e simetria, associadas à distribuição de variáveis aleatórias. Estas medidas foram selecionadas de um conjunto de alternativas por apresentarem uma dispersão relativamente maior.

Jenke, Peer e Buss [25] propõem um método de seleção de elétrodos a utilizar e de características baseado em effect-size. Isto irá obter o conjunto de elétrodos e características que melhor permitem distinguir entre estados emocionais. É utilizado um Classificador Quadrático para classificação. Isto permite separar conjuntos de medidas sobre uma superfície quadrática, permitindo distinguir entre duas ou mais classes. A utilização de uma superfície quadrática permite obter uma curva de separação de dados mais complexa que uma superfície linear.

A seleção intra-indivíduo revela uma taxa de decisões corretas de 57,5%. No entanto, quando considerada a seleção inter-indivíduo (extração das melhores conjuntos de elétrodos e características tendo em conta todas as medições realizadas), são obtidos resultados que superam os métodos clássicos de seleção utilizados.

Em 2014, Jenke, Peer e Buss [26] apresentam um artigo de revisão e teste de métodos comumente utilizados para seleção de características e elétrodos a utilizar. Os resultados demonstraram uma performance superior de métodos avançados de extração de características sobre a utilização do espectro de potência dividido por diferentes bandas. Também é demonstrada uma preferência dos métodos testados por elétrodos sobre o lóbulo parietal (responsável por integrar informação sensorial) e centro-parietal.

Candra et al [6] sugerem a divisão dos sinais em frequência por diferentes bandas previamente à extração de características. É utilizada a energia e a entropia do sinal transformado (Wavelet) como características a classificar. Para a classificação foram utilizadas

máquinas de vetores de suporte. O método pretende classificar tanto o valor de prazer como de excitação. Após isto, a combinação destes dois valores irá resultar numa emoção concreta. Foi obtida uma taxa de classificações corretas de 77.4%.

2.3.3.3 Outras Fontes

Rong, Lin e Chen [40] pretendiam testar a hipótese de identificar diferentes emoções manifestadas na fala de uma pessoa. Utilizaram dados de áudio de participantes chineses de uma série de bases de dados. É utilizado um método conhecido de criar diferentes árvores de decisão a operar sobre as características dos dados a analisar. Uma análise coletiva das decisões de cada árvore permite determinar a decisão final.

O algoritmo revelou uma performance superior relativamente aos algoritmos normalmente utilizados ⁵.

Gunes e Pantic [20] publicaram em 2010 uma análise das tecnologias modernas (para a altura) de reconhecimento emocional automático e contínuo. O artigo explora as diferentes fontes de informação emocional assim como os diferentes métodos de a recolher. Explorar principalmente a questão da dimensionalidade dos dados a classificar, referindo que estes normalmente são dotados de um grande conjunto de características. A estratégia que é referida como sendo a mais utilizada para o reconhecimento automático e “contínuo” é reduzir o problema de reconhecimento a um problema menor. Por exemplo, em vez das seis emoções, utilizar quatro. Isto pode ser feito simplesmente considerando menos pontos (utilizar domínios mais pequenos por dimensão como, por exemplo, usar só dois extremos e um neutro). Convém referir que o reconhecimento de emoções automático estava na fase pioneira aquando a escrita do artigo, havendo muita investigação foi feita na área até aos dias de hoje. Contudo, é sempre necessário ter uma ideia de onde a investigação nesta área (reconhecimento contínuo) começou.

Em 2011 Gunes e Pantic juntam-se a Nicolau [36] para testar duas aproximações clássicas para a classificação em tempo contínuo. Mais uma vez (relativamente ao artigo anteriormente referido) é referida a necessidade de se extrair conclusões em ambientes naturais e dinâmicos.

Pretende-se realizar o reconhecimento no plano prazer-excitação continuamente. Para tal foi utilizada informação proveniente de diferentes fontes. Foram recolhidos dados experimentais de sujeitos provenientes de informação sonora, movimento de ombros e expressões faciais.

O processamento foi realizado através de dois métodos. No primeiro método foi utilizada uma Rede Neuronal de Longa Memória de Curto Prazo. A diferença principal

⁵Os métodos comumente utilizados nestes casos são análise em componentes principais em paralelo com escalamento multidimensional. A ACP [38] permite, através de uma transformada ortogonal, converter um conjunto de dados que podem ou não estar correlacionados, num conjunto de dados não correlacionados. Estes são ordenados por ordem de variância, de forma a que se saiba qual é a componente que terá maior responsabilidade pela variância dos dados. EMD [11] é um método que permite medir o grau de semelhança de indivíduos dentro de um set de dados. Os dados são dispersos num espaço N-dimensional, de forma a que se tenha uma noção de distancia entre indivíduos.

para redes neuronais comuns é o facto de estar adaptada para classificação quando existem intervalos desconhecidos entre eventos relevantes. No segundo método foi utilizada uma máquina de vetores de suporte para regressão. Os resultados demonstram que o método consegue reproduzir grande parte dos valores de prazer e excitação indicados por colaboradores para os dados utilizados.

2.4 Conclusões

Foi encontrado um certo reconforto ao procurar literatura semelhante. Conclui-se que grande parte da investigação feita falha nos seguintes aspetos:

- Incoerência na nomenclatura utilizada para descrever emoções, tornando difícil a compreensão dos objetivos da investigação e a interpretação por outros investigadores;
- Testes pouco ambiciosos, realizados sobre poucos colaboradores;
- Taxas de sucesso relativamente baixas, regra geral sobre poucos sujeitos experimentais, pelo que se podem afirmar imediatamente não significativos.

Em adição a isto, não foi encontrada investigação específica sobre o problema a abordar - transmissão de uma mensagem através de um canal humano, no qual sofre forte "modelação". Não se está à procura da forma da mensagem que foi enviada, mas sim de uma repercussão da mesma, com forma desconhecida.

Grande parte da literatura referente à transmissão de mensagens num canal humano pretende captar exatamente o mesmo sinal que foi transmitido, numa outra área do corpo.

FUNDAMENTOS TEORICOS

“Every kind of science, if it has only reached a certain degree of maturity, automatically becomes a part of mathematics.”, David Hilbert [28]

A ideia deste capítulo passa por explicar o raciocínio por trás da construção do pipeline de processamento. São explicados os métodos matemáticos utilizados assim como a sua origem.

Como foi explicado no capítulo 1, a detecção dos níveis IAPS (i.e. agrado) é considerada como um problema de detecção. Como tal, a teoria subjacente ao método utilizado provém da teoria do filtro adaptado [48].

A teoria do filtro adaptado pressupõe a detecção de um sinal conhecido (e.g. $s[n]$) corrompido por ruído. Isto é matematicamente concretizado considerando que se pretende maximizar o rácio entre a potência do sinal conhecido e a potência do ruído à saída do filtro.

Como consequência da restrição da maximização do rácio sinal ruído à saída do filtro, provém que os parâmetros do filtro devem corresponder aos parâmetros do sinal esperado, invertido no tempo.

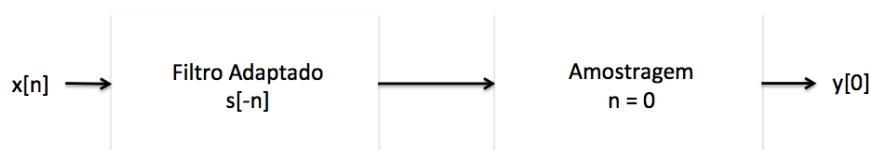


Figura 3.1: Diagrama representativo da aplicação de um filtro adaptado sobre o sinal observado.

Ao amostrar a saída do filtro adaptado no instante inicial, a saída do diagrama 3.1 torna-se a correlação entre o sinal observado e o sinal esperado.

Quanto à detecção da presença do sinal esperado, a teoria convencional dita que se estabelece um limiar acima do qual se considera que se está na presença do sinal. No entanto, no caso desta dissertação, pretende-se detetar a presença de um de oito sinais possíveis diferentes.

Isto pressupõe a utilização de vários filtros adaptados (um por sinal esperado). A detecção do sinal presente é feita verificando o máximo de todas as saídas de todos os filtros aquando o início de cada slide.

Este método foi desconstruído e especificado para o caso desta dissertação.

3.1 Descrição formal do problema

Do processo de aquisição de dados advém o vetor de valores de tensão ao longo do tempo:

$$\mathbf{v}_{\text{exp}} = [a_0, a_1, \dots, a_{T_{\text{exp}}}] \quad (3.1)$$

Este sinal é recortado em segmentos, denominados de s_s . Haverá tantos segmentos quanto símbolos na mensagem. Um segmento é um intervalo, localizado no tempo relativamente ao início de cada slide - T_s .

É adotado um tempo standard para todos os segmentos e, consequentemente, todos os estimadores - T . Este tempo depende do processamento de dados a realizar: depende de quantas amostras se está a considerar antes e depois do início de cada slide. Estes dois parâmetros (número de amostras antes - T_a e depois - T_d do início do slide - T_s) têm um elevado impacto nos resultados finais, pelo que para a mesma experiência são testados diferentes valores dos mesmos, como se poderá verificar no capítulo 6.

Pode-se definir cada um dos 63 segmentos temporais:

$$\mathbf{s} = [a_{T_s-T_a}, a_{T_s-T_a+1}, \dots, a_{T_s-1}, a_{T_s}, a_{T_s+1}, \dots, a_{T_s+T_d}] \quad (3.2)$$

Cada um destes segmentos terá um nível IAPS associado. Pode-se definir então um vetor, que represente todas as ocorrências de um sinal de determinado nível $V_i, i \in \{1, 2, 3, 4, 6, 7, 8, 9\}$:

$$\mathbf{S}_i = [s_{i,1}, s_{i,2}, \dots, s_{i,K}] \quad (3.3)$$

E, por sua vez podemos definir cada um dos seus elementos:

$$\mathbf{s}_{i,k} = [\alpha_1, \alpha_2, \dots, \alpha_T] \quad (3.4)$$

Pode-se definir também cada valor de tensão por amostra - $s_{i,k}[n] = \alpha_n$. Cada um destes segmentos pode ser dividido em duas componentes: uma pura - r - e uma

ruidosa - η . Por componente pura entenda-se o sinal puro, que ocorre sempre aquando a mesma entrada (i.e. mesmo nível). Quanto à componente ruidosa, entenda-se todo o sinal que não está associado à resposta ao estímulo. Ruído serão todos os outros sinais que não surgem como consequência da estimulação, pelo que a sua forma não é igual em todos os segmentos de igual nível. O ruído é assumido como sendo ruído gaussiano de média nula e variância constante.

Pode-se definir cada segmento $s_{i,k}[n] = r_i[n] + \eta_k[n]$

O que é pretendido descrever com esta definição da observação - $s_{i,k}[n]$ - é que a observação realizada é consequência de dois fatores: um fator inerente à observação em si - o ruído η_n - e um fator inerente ao símbolo associado à observação - o sinal puro r_i . Esta aproximação pressupõe que todos os estímulos do mesmo nível sejam considerados como estímulos equivalentes - no sentido de provocarem respostas equivalentes do sistema. Isto pressupõe que o sistema seja estacionário durante o intervalo em que a experiência é realizada. Ou seja, estímulos equivalentes irão gerar as mesmas respostas em diferentes intervalos de tempo. No entanto, o sinal em si virá adicionado de diferentes valores de ruído, pelo que não será o mesmo em todos os intervalos de tempo. O efeito do ruído pode ser diminuído, como se descreve na secção 3.3.1.1.

3.2 Estimulo

É necessária uma descrição específica do estímulo. No início começa-se com símbolos. É gerada uma sequência aleatória utilizando um alfabeto relativamente simples.

Como alfabeto inicial serão utilizados todos os número de 1 a 9, exceto 5 - não se pretende estimar o valor neutro e, pode-se, desta forma, ter um número par e um conjunto de valores diferentes que pode ser representado por três bits. Foi utilizada uma forma simplificada de gerar estas sequências - utiliza-se a teoria referente a sequências aleatórias binárias para gerar três sequências diferentes. Estas sequências são utilizadas para gerar, um numero de três bits, utilizando um elemento de cada sequência. Após este processo obtém-se uma sequência de 63 valores, podendo cada um assumir um de 8 níveis IAPS diferentes.

3.2.1 Sequências Maximamente Aleatórias

Sequências maximamente aleatórias (m-sequências) são sequências binárias pseudo-aleatórias. Uma m-sequência é gerada através de um polinómio mínimo associado a um anel polinomial sobre 2^n - uma vez que se se pretende obter sequências binárias.

Para gerar sequências binárias pode ser utilizado um Linear-FeedBack-Shift-Register (LFSR). Basicamente, um LFSR é um shift register que altera o seu estado segundo uma função linear. Uma m-sequência pode ser gerada a partir de um LFSR em que a função escolhida esteja associada ao polinómio mínimo [18, 19]. Para uma sequência de tamanho

63 é necessário um polinómio de ordem $6 - 2^6 - 1 = 63$. Para ordem 6 o polinómio mínimo é $x^6 + x + 1$ [45]. Por tamanho da sequência entenda-se o número de valores diferentes possíveis antes que a sequência comece a repetir. Na figura 3.2 pode ser vista a realização do polinómio de ordem 6 em LFSR:

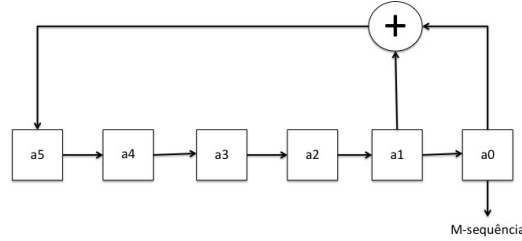


Figura 3.2: Realização do polinómio $x^6 + x + 1$ em LFSR

Cada registo comporta-se da seguinte forma:

$$\begin{cases} a_5[n+1] = a_0[n] + a_1[n], \\ a_4[n+1] = a_5[n], \\ a_3[n+1] = a_4[n], \\ a_2[n+1] = a_3[n], \\ a_1[n+1] = a_2[n], \\ a_0[n+1] = a_1[n]. \end{cases}$$

Uma característica típica de m-sequências é a sua auto correlação circular ser um impulso. Isto implica um valor alto para $n = 0$ e aproximadamente $-\frac{1}{N}$ para $n \neq 0$ [42], em que N representa o tamanho da sequência. É possível criar até 64 sequências diferentes com um polinómio de ordem 6. Isto é feito considerando os 64 valores diferentes para a semente ¹ da sequência.

No entanto a sequência criada pela semente nula $[0\ 0\ 0\ 0\ 0\ 0]$ não irá alterar o seu estado para a função apresentada. Na prática teremos apenas 63 sequências possíveis.

Começar com uma semente diferente quer dizer, em outras palavras, que a sequência começa num ponto diferente, podendo ainda transitar de estado 63 (i.e. $2^{\text{Ordem do Polinómio}} - 1$) vezes até que a sequência volte ao ponto onde começou. Portanto, da auto correlação circular de uma m-sequência ser um impulso pode-se deduzir que a correlação de uma m-sequência com outra m-sequência da mesma ordem mas semente inicial diferente será aproximadamente 0.

¹semente - valor inicial de cada um dos registos. Neste caso, o valor inicial- 0 ou 1 - de cada um dos 6 bits

3.3 Modelo

3.3.1 Estimação de Máxima Verosimilhança

Estimação de máxima verosimilhança [22] pressupõe a observação repetida de um determinado valor corrompido por ruído - $s_k[n] = s[n] + \eta_k[n]$ - para K observações do valor $s[n]$. O valor a estimar mantém-se constante ao longo das observações, enquanto que o valor do ruído que o corrompe é variável.

Para o caso de ruído gaussiano branco aditivo o valor a estimar concretiza-se na média de todas as observações realizadas.

3.3.1.1 Repetição de Medidas

Sendo S a potência do sinal de tamanho N, esta define-se de forma simplificada como se segue:

$$S(x[n]) = \frac{1}{N} \sum_{n=1}^N x[n]^2. \quad (3.5)$$

A variância de um sinal de média nula é equivalente à sua potência:

$$Var(\eta[n]) = \sigma_\eta^2 = \frac{1}{N} \sum_{n=1}^N [\eta[n] - \mu_\eta]^2 = \frac{1}{N} \sum_{n=1}^N \eta[n]^2. \quad (3.6)$$

Assumindo o ruído $\eta[n]$ como tendo media nula, $\mu_\eta = 0$. Obtém-se então:

$$SNR = \frac{S(x[n])}{\sigma_\eta^2}. \quad (3.7)$$

Após K observações, pode-se calcular o valor médio do sinal de nível i - $s_{i,Avg}$:

$$s_{i,Avg} = \frac{1}{K} \sum_{k=1}^K s_{i,k} = \frac{1}{K} \sum_{k=1}^K r_i + \frac{1}{K} \sum_{k=1}^K \eta_k = \frac{K}{K} r_i + \frac{1}{K} \sum_{k=1}^K \eta_k. \quad (3.8)$$

Deduz-se então que $r_i = r_{i,Avg}$, pelo que $S(r_i) = S(r_{i,Avg})$. Quanto à variância da componente ruidosa:

$$Var\left(\frac{1}{K} \sum_{k=1}^K \eta_k\right) = \frac{1}{K^2} Var\left(\sum_{k=1}^K \eta_k\right) = \frac{1}{K^2} \sum_{k=1}^K Var(\eta_k) = \frac{K}{K^2} \sigma_\eta^2 = \frac{\sigma_\eta^2}{K}. \quad (3.9)$$

Deduz-se por fim que a relação de sinal ruído para múltiplas observações toma a forma:

$$SNR_{Avg} = K \frac{S(r_i[n])}{\sigma_\eta^2} = K \times SNR. \quad (3.10)$$

Está comprovado que a relação sinal ruído varia proporcionalmente ao número de observações.

É de referir que estes resultados não são desconhecidos aos investigadores do P300. O potencial P300 (assim como outros potenciais dignos de estudo) vem corrompido de ruído com amplitude quase que igual à do sinal. Calcular a média de várias reações a estímulos semelhantes permite isolar o sinal [14].

3.3.1.2 Estimadores

Os estimadores de cada sinal puro, de cada nível IAPS, r_i serão a média de todas as suas K observações $S_i : E[S_i]$.

$$\hat{r}_i = \frac{1}{K} \sum_{k=1}^K s_{i,k}. \quad (3.11)$$

Isto é devido ao facto de não se possuir nenhuma descrição de função de distribuição de probabilidade - PDF - do valor que se quer estimar. Contudo, considera-se que o ruído é gaussiano de média nula, o que permite utilizar a estimação por máxima verosimilhança, que, neste caso, é a média de todas as observações realizadas [22].

3.4 Deteção

3.4.1 Votação

De forma a tomar uma decisão sobre as 8 possibilidades diferentes, é necessário utilizar um valor que seja representativo da semelhança entre cada segmento e os 8 estimadores.

3.4.1.1 Correlação Cruzada e Decisão

A relação de semelhança entre cada segmento e cada um dos 8 estimadores, para cada canal, será dada pela correlação cruzada com 0 amostras de atraso:

$$c(s, i) = \frac{\sum_{n=1}^N (s_s[n] - \bar{s}_s)(\hat{r}_i[n] - \bar{\hat{r}}_i)}{\sqrt{\sum_{n=1}^N (s_s[n] - \bar{s}_s)^2} \sqrt{\sum_{n=1}^N (\hat{r}_i[n] - \bar{\hat{r}}_i)^2}}. \quad (3.12)$$

O coeficiente, nestes caso, está apresentado para um determinado segmento s_s , respetivo ao slide s , e um determinado estimador \hat{r}_i de nível i . Numa visão geométrica, este coeficiente refere-se ao $\cos(\alpha)$ - cosseno do ângulo formado pelos dois vetores, neste caso, s_s e \hat{r}_i [19].

É de elevada importância centrar os estimadores e os segmentos (i.e. subtrair a média - respetivamente $\bar{\hat{r}}_i$ e \bar{s}_s). Caso não estejam centrados apenas se pode verificar a ortogonalidade entre os vetores. A não correlação entre variáveis apenas pode ser verificada caso as variáveis estejam centradas [39].

Ao reduzir os vetores à energia unitária garante-se que o valor da correlação entre ambos

dê um valor entre 0 e 1.

Ambos estes passos estão considerados na equação 3.12 ².

A decisão sobre o nível é tomada, para cada segmento s , como sendo o valor i (i.e. nível IAPS) que maximiza:

$$\max_{1 \leq i \leq 9, i \neq 5} (c(s, i)). \quad (3.13)$$

²No entanto, aquando a implementação, este passo de normalização apenas deve ser executado uma vez para cada vetor de dados (i.e. estimadores e segmentos) de forma a poupar em tempo de processamento desnecessário.

MÉTODO EXPERIMENTAL

Nesta secção é descrito o método experimental. O método está dividido em 5 partes. Os requisitos pré-experimentais configuram o material que tem que estar disponível de forma a que se possa realizar a experiência. A distribuição de tarefas refere-se às entidades que devem estar presentes no decorrer da experiência e às tarefas que cada entidade está destinada a executar. O cenário experimental descreve como o espaço onde decorre a experiência deve estar organizado de forma a garantir que esta pode ser executada. O procedimento experimental descreve passo a passo o conjunto de tarefas, por ordem, que devem ser executadas. A realização do processamento descreve o conjunto de intervenções humanas que devem ser executadas sobre os dados até que estes possam ser trabalhados pelo software criado.

4.1 Requisitos Pré-experimentais

Antes da realização da experiência é necessário que os seguintes itens estejam disponíveis e prontos a utilizar:

- Touca onde colocar os elétrodos;
- 32 elétrodos EEG + 8 elétrodos periféricos;
- Pelo menos uma bateria carregada;
- Mapa com o local onde serão colocados os elétrodos;
- Lista de correspondência entre o nome dos elétrodos na touca e no software;
- Slide-show e respetiva mensagem associada;
- Pipeline de processamento;

- Guia para a realização da experiência;
- Sistema de aquisição de dados;
- Material de limpeza: escovilhão e escova.

4.2 Distribuição das Tarefas

Haverá três funções distintas que deverão ser distribuídas idealmente por 3 pessoas, mas que poderão ser realizadas todas pela mesma pessoa:

- Realizador - Deverá estar constantemente a par de tudo o que se passar durante o realizar da experiência, devendo portanto ser informado de cada passo. Deverá ajudar o técnico de operações durante tarefas que possam ser realizadas em paralelo. Tem que dar indicações sobre todos os passos da experiência, comunicando com os outros dois colegas. Tem que ser capaz de procurar retificar eventuais falhas no equipamento.
- Técnico de Operações - Deverá preparar o participante para a aquisição. Deverá executar o registo da aquisição, informando o realizador do decorrer da mesma.
- Curador Deverá zelar pelo equipamento, certificando-se que este é manipulado corretamente.

Caso apenas hajam duas pessoas, é sugerido que as tarefas de Curador e Realizador sejam atribuídas à mesma pessoa.

4.3 Cenário Experimental

Cerca de uma hora antes de cada aquisição de dados deverá verificar-se a sala para conferir que o cenário se encontra como descrito de seguida. Esta verificação deverá ser feita pelo Curador:

- Dossier com declarações de consentimento fora da sala onde será realizada a experiência;
- Computador de aquisição de dados fora da sala, ligado à corrente;
- Duas baterias têm que estar disponíveis e pelo menos uma delas deve estar completamente carregada. A bateria selecionada para a aquisição deverá estar debaixo do sistema de aquisição;
- Sistema de aquisição de dados em cima de uma mesa, atrás da cadeira onde o participante estará sentado;

- Bus de 32 elétrodos e os 8 fios dos elétrodos externos enrolados num estendal, destinado apenas à arrumação dos cabos;
- Cadeira em frente ao computador destinado a mostrar o slide-show;
- Computador para mostrar slide-show, com bateria carregada, ligado à corrente, com os ficheiros .pdf para apresentação;
- Ambos os computadores - de aquisição e de slide-show - ligados;
- Aplicador de gel cheio;
- Fibra ótica suspensa e ligada ao computador de aquisição de dados.

4.4 Procedimento Experimental

A experiência deverá ser realizada numa Gaiola de Faraday, com as luzes apagadas, e a porta fechada, de forma a minimizar os efeitos do ruído nos dados adquiridos. Apenas o sujeito está na sala durante a gravação dos dados adquiridos.

O método experimental é descrito de seguida, ao pormenor, desde a chegada do participante até à conclusão da aquisição. As tarefas estão divididas pelas três funções - Realizador (R), Técnico de Operações (T), Curador (C):

1. R - O participante entra na sala e preenche a declaração de consentimento (ver anexo B);
 - De forma a justificar a obtenção dos dados do participante, é necessário um documento legal que comprove que a aquisição dos mesmos foi voluntária.
2. R - Breve explicação da experiência e do âmbito da mesma ao participante;
 - A explicação terá que ser breve uma vez que o participante não deve estar consciente do conteúdo do slide-show que vai ver. No entanto é avisado que algumas imagens podem ser chocantes.
3. TO - O participante entra na sala e senta-se na cadeira em frente ao computador onde será mostrado o slide-show;
4. TO - É confirmada que a distância desde os olhos até ao ecrã corresponde à distância definida como padrão;
 - A distância é de 1 metro, de forma a garantir que todos os participantes estarão em condições o mais semelhantes possíveis. A distância é garantida por meio de um cabo com tamanho de aproximadamente 1 metro.
5. TO - É colocada a touca na cabeça do participante;

- A toca contém 128 entradas para eléttodos, das quais apenas 32 são utilizadas.
6. TO - É colocado gel condutor na entrada correspondente a cada eléttodo a colocar, seguido da colocação do mesmo nas posições indicadas na figura 4.1;
- O gel permite reduzir a resistência entre o escalpe e a ponta do eléttodo, aumentando desta forma a intensidade do sinal recebido. Após colocados os 32 eléttodos são colocados os 8 eléttodos EX1, ..., EX8 à superfície da pele:
 - a) EX1 e EX5 - Sobre os olhos (esquerdo e direito respetivamente), alinhados com o centro da pupila.
 - b) EX2 e EX6 - Ao lado dos olhos (esquerdo e direito respetivamente), entre os olhos e as orelhas, alinhados com o centro da pupila.
 - c) EX3 e EX7 - Por baixo dos olhos (esquerdo e direito respetivamente), alinhados com o centro da pupila.
 - d) EX4 e EX8 - Na nuca, do lado esquerdo e direito respetivamente, a distâncias equivalentes da coluna.
7. C - Após colocados todos os eléttodos nas entradas previamente definidas , montar o equipamento de aquisição de dados;
- a) Ligar bateria ao sistema de aquisição de dados;
 - b) Ligar bus de 32 eléttodos;
 - c) Ligar os 8 eléttodos individuais;
 - d) Ligar fibra ótica.
8. C - Ligar o equipamento de aquisição de dados;
- Ao ligar o equipamento, e possível verificar se os eléttodos de referencia (CMS e DLR) estão a fazer contacto. Se a luz do “CM in Range” estiver a piscar é necessário retificar a ligação dos mesmo, colocando mais gel. A luz deixará de piscar, permanecendo azul se os eléttodos de referencia estiverem bem ligados.
9. R - Tirar uma foto com a identificação(matricula) da experiência visível;
- São na verdade tiradas 4 fotos. Uma de cada lado, uma de frente e uma de trás. As fotos devem apanhar todos os eléttodos e a identificação da experiência. O principal objetivo destas fotos é o de no futuro poder verificar como estava a montagem da experiência.
10. R - Verificar se as ligações estão bem feitas, realizando uma observação prévia dos sinais;
- É também possível verificar o offset de todos os eléttodos, para se certificar que nenhum está a fazer mau contacto.

11. TO - Se o sinal correspondente a algum elétrico não for consistente com o esperado, acrescentar mais gel à entrada correspondente ao mesmo;
12. TO - Colocar a placa a tapar o ecrã (i.e. teste de controlo) de forma a confirmar que não existe relação entre possíveis inferências sobre os dados de controlo e as inferências que se tentará realizar sobre os dados da segunda parte da experiência;
13. TO - Colocar o software em modo de gravação;
 - Após clicar no botão “play” basta carregar no botão “save file” de forma a colocar o software em modo gravação. No nome do ficheiro deve ser colocado a identificação da experiência. Na descrição do ficheiro deve vir a temperatura e a humidade relativa.
14. R - Desligar o quadro elétrico, de forma a minimizar a interferência;
15. TO - Colocar a gaiola em ground;
16. TO - Iniciar o slide-show;
17. TO - Sair da sala;
18. TO - Verificar o fim do slide-show através das comutações dos transístores;
 - É possível verificar no software de aquisição de dados a transição dos valores lidos pelas duas entradas correspondentes aos dois transístores. Quando estas deixarem de comutar, passados aproximadamente 4 segundos, o slide-show terminou (no entanto deve ser esperado mais tempo que 4 segundos, caso apenas tenha havido uma falha nos transístores, se bem que é improvável que uma falha se dê nos dois transístores ao mesmo tempo).
19. TO - Retirar a placa da frente do ecrã;
20. TO - Reiniciar o slide-show;
21. TO - Sair da sala;
22. TO - Verificar o fim do slide-show através das comutações dos transístores;
23. TO - Desativar o modo de gravação;
 - Para isto basta apenas clicar no botão de "stop save", deixando o software no play;
24. TO - Desligar a gaiola do ground;
25. R - Ligar o quadro elétrico;
26. C - Desligar o equipamento de aquisição de dados;

27. TO - Retirar os elétrodos da toca;
28. TO - Retirar a touca do sujeito;
29. C - Proceder à limpeza do material utilizado;
30. TO - Transferir os dados para uma pen;
 - Posteriormente estes dados devem ser transferidos para o computador de processamento de dados.
31. C - Colocar todo o material onde este se encontrava antes de se realizar a experiência;
 - Isto implica deixar todo o material onde este se encontrava e deixar todo o equipamento ligado ou desligado, dependendo de como foram encontrados na sala:
 - Touca sobre a cabeça de manequim;
 - Computadores desligados;
 - Extensões desligadas;
 - Cabos (bus e 8 elétrodos individuais) enrolados em torno do estendal;
 - Bateria utilizada a carregar.

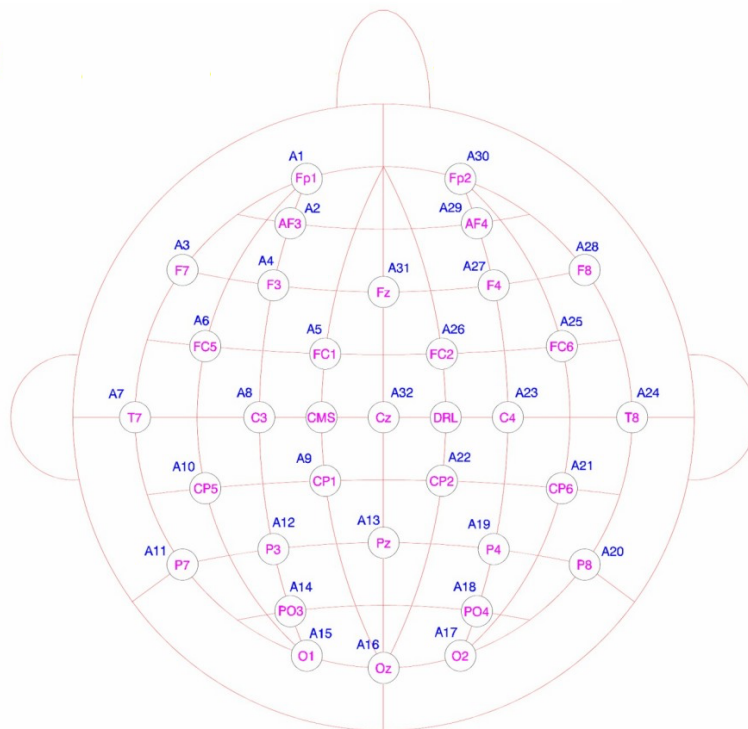


Figura 4.1: Posicionamento dos 32 elétrodos + 2 elétrodos de referencia CMS e DRL [10]

A touca e os elétrodos devem ser limpos apenas com água e com uma escova de dentes e um (perguntar o nome do fio dentário metálico) para limpar os elétrodos e a touca respetivamente.

4.5 Realização do Processamento

De forma a realizar o processamento dos dados, os seguintes passos devem ser seguidos pela ordem indicada::

1. Os ficheiros que vêm diretamente do software de aquisição de dados (em formato .bdf - BioSemi Data Format) têm que ser convertidos num formato compatível com o software de processamento (.mat - ficheiros de dados Matlab ®) - deverá ser utilizado o software eeglab(disponível online, no site do Matlab ®) para conversão dos dados (de formatos típicos de ficheiros de dados de eletroencefalogramas);
2. Do passo anterior proveem - a matriz $40 \times n^{\circ} \text{Amostras}$ correspondente aos sinais gravados dos 32 elétrodos mais 8 canais externos e os dados lidos de outras entradas, mais especificamente, relativas aos foto transístores;
3. Criar uma pasta nova com o nome da matrícula da experiência;
4. Na pasta introduzir a versão mais recente do pipeline de processamento e as funções para visualização dos dados. Introduzir também a pasta "DataInicial" e mudar o nome para "Data". Na pasta encontra-se o ficheiro com os dados dos slides - kEpoch. Encontra-se também a pasta eeg, onde deve ser colocado o ficheiro .mat convertido do ficheiro .bdf;
5. A partir deste passo o trabalho é realizado pelo software de processamento, requerendo alguma visualização por parte de quem executar o pipeline de forma a verificar que toda a informação foi extraída corretamente, e, caso contrário, eliminar erros dos dados;
6. É necessário verificar se houve tantas transições do sinal dos foto transístores como slides e se estas transições de sinais se encontram no momento certo;
7. Após a verificação o pipeline deverá ser corrido, sem interrupções, para diferentes tamanhos da janela, o que configura o tamanho dos segmentos/estimadores. Também deverão ser considerados diferentes intervalos de frequência;
8. No final deverá recorrer-se às funções criadas para visualização dos resultados obtidos.

IMPLEMENTAÇÃO

Neste capítulo será discutida a implementação na prática do projeto proposto. Inicia-se pelo slide-show criado, explicando a sua construção. É explicado de seguida como se garante o conhecimento do momento de transição entre slides. É apresentado o cenário experimental e o equipamento usado. Por fim explica-se o código criado para processar os dados, assim como algumas considerações de organização e de nomenclatura. A explicação do pipeline está dividida consoante as diferentes etapas do mesmo.

Quanto ao pipeline, cada etapa corresponde a um módulo. Ao tratar de dados o software deverá estar dividido em tarefas elementares. O objetivo é reduzir qualquer dependência entre os módulos. Obviamente existe sempre dependência em termos de informação - um módulo não poderá correr se o módulo anterior não tiver produzido os dados necessários. No entanto, existe independência no sentido de cada módulo poder correr isolado. É possível correr um módulo, corrigir o código do mesmo e voltar a correr o mesmo módulo sem necessitar de alterar os módulos anteriores. É também boa filosofia de programação que se possa explicar o funcionamento de um módulo sem se precisar de explicar o funcionamento de todos os outros [51].

5.1 Slide-Show

Existem três tipos de slides:

- Slides de transição - Estes slides servem de auxílio. O começo de um slide (que não seja de transição) é detetado pelos foto-transístores como uma mudança de valor. Dessa forma, estes slides estão destinados a baixarem o valor lido pelos foto-transístores até ao mínimo possível. Apenas estes slides têm os cantos pretos. Todos os outros têm os cantos brancos, de forma a tornar máximo o valor lido pelos foto-transístores.

- Slides de “refresh” - Estes slides estão destinados a dar um momento para o participante movimentar os olhos antes de começar a ver as imagens IAPS. São slides de fundo cinzento. Contêm uma circunferência colorida que pode estar em diversos pontos do slide.
- Slides de imagem IAPS - Estes slides estão completamente preenchidos, cada um com uma imagem IAPS.
- Slides de imagem não classificada - Estes slides estão completamente preenchidos com uma imagem sem classificação prévia.

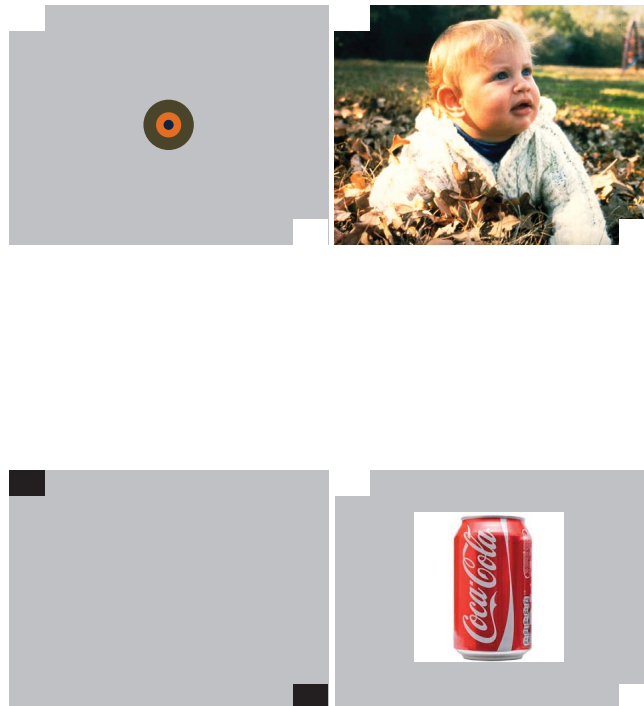


Figura 5.1: Slides exemplificativos de “refresh” - canto superior esquerdo -, imagem IAPS - canto superior direito -, transição - canto inferior esquerdo - e imagem não classificada - canto inferior direito.

O slide-show começará com um slide de transição. Após isto são mostrados um conjunto de slides de refresh em que as circunferências surgem em diferentes pontos. Isto sugere ao participante que acompanhe o movimento das circunferências - chamar-se-á a este conjunto de slides de conjunto de refresh.

De seguida são apresentados os slides de imagens IAPS/imagem sem classificação. O slide-show deverá conter 63 imagens IAPS (mensagem com 63 símbolos) e 8 imagens não classificadas. Ao todos o slide-show deverá ter 71 imagens. Cada imagem aparece seguida de um slide de transição. Isto permite criar uma transição nos valores lidos pelos foto-transístores - o que, consequentemente, irá permitir a deteção dos inícios dos slides de imagem.

Os slides de imagens de marca são inseridos ao longo do slide show sem nenhum critério em particular. Convém, no entanto, que não sejam mostradas duas imagens de marca de seguida. Deverá ser mostrada a primeira metade (35 imagens + 35 transições) do slide-show seguida de uma nova sequência de refresh. Após a sequência de refresh deve ser mostrada a segunda metade (36 imagens + 35 transições).

Para as temporizações de transição de slides foram escolhidos números primos. Slides de imagem devem durar 2,63 segundos, enquanto que slides de transição e de refresh devem durar 0,43 segundos.

5.2 Fototransistores

5.2.1 Sincronização Temporal

Ao utilizar uma frequência de amostragem elevada para aquisição de dados, está-se a fazer um compromisso de precisão no tempo. Como se está a gravar tantas amostras por segundo, se houver alguma diferença entre o tempo onde o estímulo deveria começar e onde efetivamente começou, esta diferença pode-se traduzir num numero elevado de amostras. Esta diferença irá introduzir uma componente de erro adicional em diferentes pontos do processo - os segmentos não correspondem aos momentos exatos de cada slide. Isto vai introduzir erros nos próprios segmentos, e, consequentemente, nos estimadores e na correlação segmento/estimador.

É necessário utilizar um método que permita controlar o momento exato de exposição de cada slide/estimulo. Para esse efeito foram colocados foto-transístores nos cantos (superior esquerdo e inferior direito - correspondentes ao canto onde começa a mudança de pixeis e onde acaba, respetivamente) do computador de exposição de slides. Isto permite aproximar o tempo em que o slide começou a mudar e o tempo em que acabou.

Às imagens foram acrescentados quadrados pretos nos cantos referidos às imagens ímpares - slides de transição - e quadrados brancos às imagens pares - slides de imagem e refresh. Este método permite detetar a transição entre slides, uma vez que dos foto-transístores virá um valor diferente quando for um slide par ou um slide ímpar. Este valor é lido pelo mesmo sistema de aquisição que integra os dados dos elétrodos.

5.3 Equipamento e Cenário

O equipamento envolve os elétrodos, o conversor analógico digital e a toca. Todo este equipamento é proveniente da empresa Biosemi. Este equipamento foi escolhido preferencialmente sobre outros de elevada precisão uma vez que vinha já com a touca - Figura 5.2.

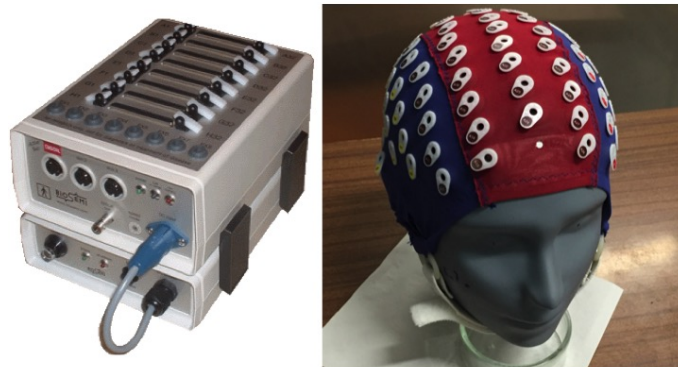


Figura 5.2: Conversor analógico digital Biosemi [3] - esquerda. Touca para colocação dos elétrodos - direita

O cenário onde o participante se senta e realiza a experiência pode ser observado na figura 5.3. No cenário pode ser vista a cadeira e o computador onde é mostrado o slide-show. No computador pode-se observar a cobertura superior e inferior que tapa o ecrã. Esta cobertura está destinada a esconder os foto-transístores, de forma a isola-los da luz exterior, melhorando o seu desempenho.

5.4 Convenções de Programação

5.4.1 Nomenclatura

Algumas convenções deveram ser tomadas ao dar nome a variáveis, funções e scripts em Matlab ®.

- Funções - As funções têm alguma liberdade de escolha de nome. No entanto a primeira palavra deve estar em minúscula. Todas as outras palavras usadas no nome da função devem começar com maiúsculas.
- Script - O script deve começar com a palavra “Faz” seguida de um número que seja representativo da posição do script no correr do pipeline. Isto deverá ser seguido de um nome que seja representativo do que o script faz.



Figura 5.3: Cenário experimental. Mais especificamente, a cadeira onde o participante se senta e o computador onde é mostrado o slide-show

- Variáveis devem começar com uma letra que descreva a pluralidade de valores da mesma e seguidas de um nome representativo:
 - Valores elementares devem começar com a letra "q";
 - Vetores de valores devem começar com a letra "k";
 - Matrizes de valores devem começar com a letra "m";
 - Strings podem começar com o nome da mesma, desde que comecem em letra minúscula.

5.4.2 Funcionalidade

Os scripts serão os módulos do pipeline, se bem que o pipeline em si será também um script. Os scripts devem correr código destinado ao processamento dos dados. São responsáveis por carregar dados existentes e processá-los para criar novos dados. Raras exceções de scripts podem também alterar valores já existentes.

Funções estarão destinadas a apresentar dados, podem realizar algum processamento, mas, no entanto, o resultado desse processamento não deverá ser guardado, apenas apresentado ou devolvido pela função. O facto de se devolver os valores a apresentar permite que as funções possam ser utilizadas por scripts para se obter um valor específico ou para aceder a informação de mais alto nível (e.g. vetores com os canais com menos erros, por cada take).

5.5 Organização dos Dados

Será importante fazer um pequeno aparte para explicar que os dados estão organizados por pastas. Os dados são criados de forma a que haja um grande número de ficheiros em um grande número de pastas. Isto permite criar dados que tenham o mínimo de valores necessários possível. Por exemplo, em vez de criar uma matriz de 820 linhas e T_{exp} colunas - valor que rondam os 8 milhões de amostras - criar 820 ficheiros, cada um correspondente a um canal. Isto permite trabalhar sobre dados mais pequenos, não sendo necessário pré-alocar matrizes grandes demais nem fazer load a dados igualmente grandes. Nas primeiras versões em que foram consideradas variáveis grandes demais, o pipeline ameaçava demorar perto de 4 dias a correr. Este tempo foi reduzido para algumas horas, após se mudar o método de organização das variáveis.

Portanto, após correr o pipeline, a pasta “Data” deverá ter o seguinte aspeto - os nomes são nomes de pastas:

1. values - nesta pasta o software guarda tudo o que sejam variáveis elementares (e.g. numero de slides, numero de níveis, numero de takes), quase que todos associados a quantidades;
2. logistic - nesta pasta vem, à partida, “kEpochs” que contém a informação relativa aos níveis de cada slide. Após isto são inseridos nesta pasta todos os dados que não contém informação em concreto, mas que serão utilizados para extrair informação ou dar algum tipo de orientação ao pipeline para as tarefas a executar (e.g. sinais dos foto transístores, matriz com informação sobre os diferentes takes a executar e os tamanhos da janela dos segmentos/estimadores respetivos);
3. eeg - nesta pasta deverão ser inseridos apenas ficheiros com informação bruta proveniente dos elétrodos. Nesta pasta vem, à partida, o ficheiro .bdf e .mat associado. É nesta pasta que é inserida a matriz com o sinal gravado de cada um dos elétrodos, extraída do ficheiro .mat proveniente do ficheiro .bdf.
4. filter - Esta pasta deverá conter apenas variáveis referentes ao filtro aplicado aos sinais. O filtro utilizado depende dos valores escolhidos para o utilizador (isto é explicado mais adiante, na secção 5.7. Também deverão ser guardadas aqui as divisões auxiliares dos 40 canais, criadas no modulo 02.5.
5. schsComb - Nesta pasta são colocadas todas as matrizes de gravação de cada um dos canais, tanto mono polares como bipolares. Terá ao todo 820 ficheiros .mat.
6. schsSegs - Esta pasta terá também 820 ficheiro .mat, mas cada um deles terá uma matriz de 71 posições por T_{seg} - ou seja, 71 posições em que cada uma irá conter um segmento correspondente a esse slide.

7. `estimateFull` - Nesta pasta são inseridos os estimadores em bruto. Os estimadores são criados, como foi explicado, realizando a média, para cada canal, dos sinais de cada nível. Isto dá um total de 820 matrizes, cada uma com 8 estimadores.
8. `estimateSpecific` - Nesta pasta são criados os estimadores que serão efetivamente utilizados. Para tal é necessário retirar o sinal referente ao segmento a analisar. Se se realizar a correlação com um estimador e o segmento a testar estiver contido no estimador, então esse nível será sempre o escolhido. Isto justifica-se pois se o sinal vier adicionado no estimador, a correlação dará sempre mais alta, sendo esse nível sempre escolhido, não havendo erro no final. Esta pasta tem um total de 820 pastas, cada uma com 71 ficheiros `.mat`, cada um com 8 estimadores. Ou seja, 8 estimadores por cada slide, por cada canal.
9. `results` - após aplicados os estimadores, é extraída a decisão do pipeline quanto a cada slide de cada canal. Esta pasta irá conter duas matrizes associadas a cada canal: uma com as decisões do software e uma com o erro relativamente ao nível real de cada canal.

As pastas `estimateFull` e `estimateSpecific` são criadas por cada vez que é executado um take, tendo um sufixo diferente por cada take (sufixo que depende dos tamanhos da janela do take). Isto justifica-se, uma vez que por cada take novo, estas pastas sofrerão alterações - ao mudar o tamanho das janelas, muda o tamanho dos segmentos e estimadores e, consecutivamente, muda os resultados obtidos. Os segmentos mudam com o tamanho dos intervalos, mas são facilmente calculados, pelo que não há necessidade de os gravar por cada take.

A pasta `results` é criada em pastas diferentes dependendo do take e da filtragem aplicada. Isto permite guardar todos os resultados para futura observação.

5.6 Pipeline

O código Matlab ® referente ao correr do pipeline e a todos os módulos utilizados pode ser observado no anexo C.

Ao dar-se o início do pipeline são criadas 3 variáveis - `qCurrentTake` e `kTakeList` `qTakes`. `qCurrentTake` refere-se ao take que está a decorrer, sendo inicializado a 1. `kTakeList` refere-se à lista que contém todos os takes a realizar - Matriz $q_{takes} \times 2$ - em que em cada linha tem os valores de cada take e na primeira coluna virá o valor “qBackWindow” e “qWindow”. Estes valores representam o quanto se deve considerar (para a extração dos segmentos) antes do início de cada slide e depois, respetivamente. Isto configurará o tamanho do intervalo/segmento.

No início do pipeline também é criada a matriz `kFilter` (referente aos diferentes intervalos de frequência), `qCurrentFilter` (referente ao filtro a utilizar) e `qFilterTakes` (referente ao número total de intervalos de frequência a considerar). Todos os tempos são guardados,

tanto o tempo que o pipeline demorou a correr, assim como o tempo que cada módulo demorou a correr.

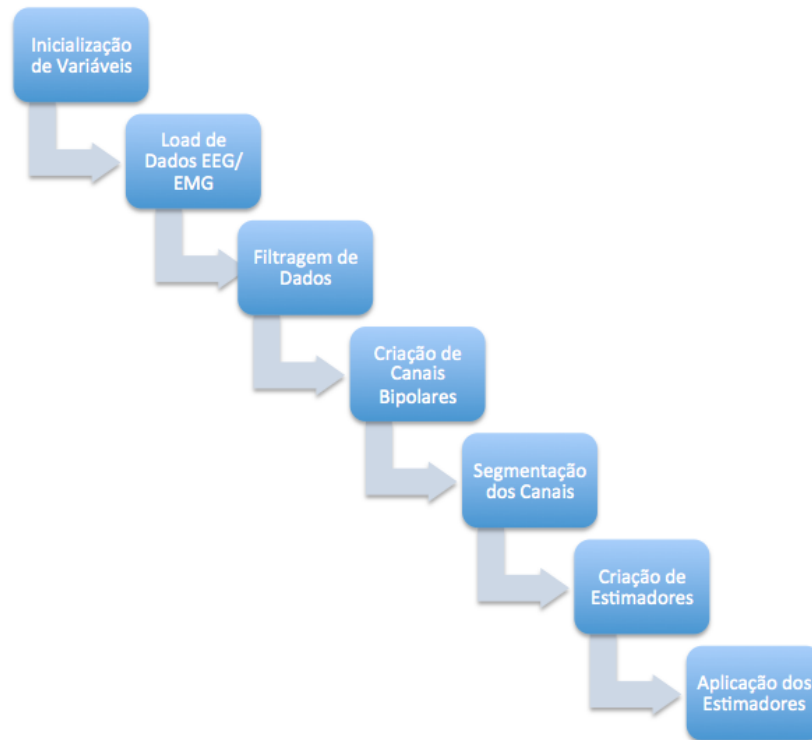


Figura 5.4: Diagrama representativo do método utilizado

A figura 5.5 demonstra o fluxo do pipeline e a figura 5.4 demonstra o fluxo do método considerado, não especificando para o caso do pipeline criado. Cada módulo será explicado de seguida. No entanto, a ideia do diagrama é dar a entender o decorrer das operações a executar sobre os dados e em que condições é que os módulos devem repetir a sua execução (caso hajam diferentes formatos dos intervalos e diferentes bandas de frequência a considerar, como será explicado de seguida).

O fluxo dos dados pode ser observado no anexo D.

5.6.1 Faz00 - Extraí Sinal Fototransistor

O primeiro módulo está responsável por extrair toda a informação possível do ficheiro .bdf que não sejam dados de leituras EEG. Este ficheiro contém informação crucial que não vem no ficheiro .mat criado pelo eeglab. É deste ficheiro que são extraídos os sinais dos fototransístor. É também deste ficheiro que é retirada a frequência de amostragem (16384 Hz).

Isto requer alguma lógica adicional. O sinal referente aos transístores, sendo uma entrada adicional, não vem incluído na conversão realizada pelo eeglab. Apenas os dados referentes a elétrodos vêm no ficheiro .mat resultante. Como tal, é necessário extrair estes

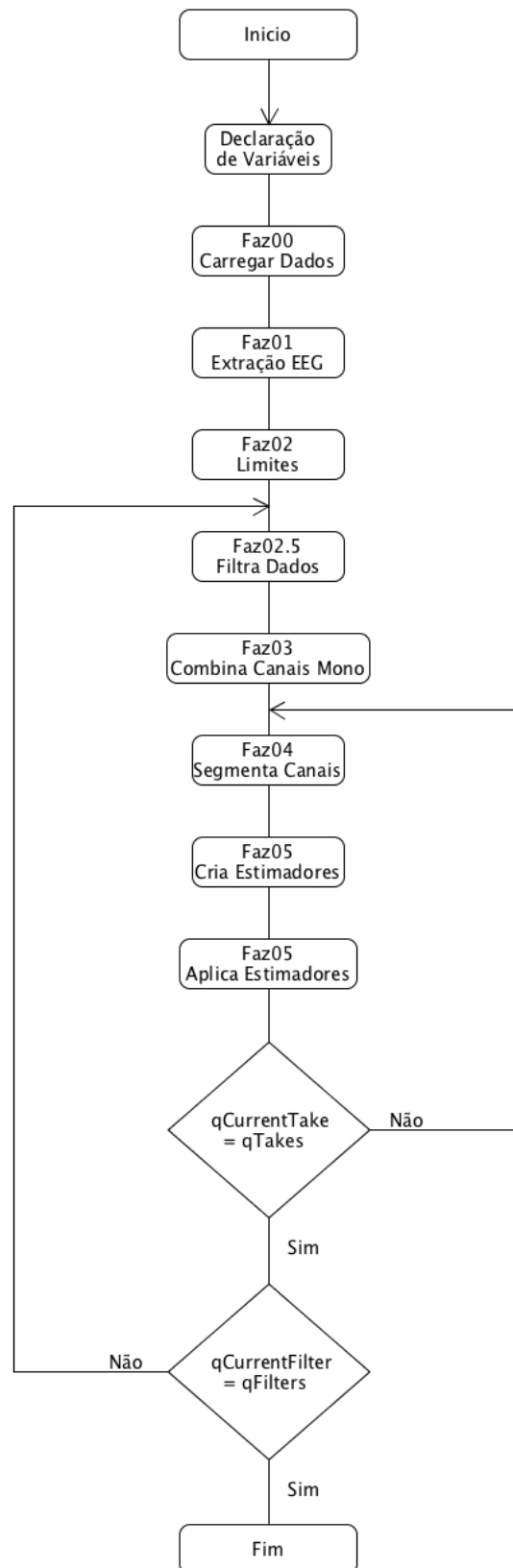


Figura 5.5: Diagrama de Execução do Pipeline

dados do ficheiro .bdf. É necessário saber em que parte do ficheiro vem a informação referente a estes canais e extrair os valores lidos para uma matriz $1 \times T_{\text{exp}}$. Neste caso resultará duas matrizes de igual dimensão, uma vez que se está a utilizar dois foto transístores. Estas duas matrizes são gravadas com os nomes kAzul (fototransístor superior esquerdo) e kVerde (fototransístor inferior direito).

Por fim é retirada a frequência de amostragem "qFs". No entanto, a frequência de amostragem pode ser retirada diretamente do ficheiro .mat.

5.6.2 Faz01 - Load Data

O ficheiro .mat resultante da conversão do ficheiro .bdf vem com demasiada informação que não é necessário utilizar. Este módulo extrai a matriz $40 \times T_{\text{exp}}$ referente à gravação da experiência inteira pelos 40 elétrodos utilizados. Esta matriz é gravada com o nome "mSchs".

5.6.3 Faz02 - Descobre Limites

Os fototransístor lêem 0 volts quando na presença de cor preta e aproximadamente 1 volts na presença de branco. Isto são os valores presentes nas matrizes kAzul e kVerde. A vantagem de utilizar dois fototransístor, além de se poder verificar se existe algum atraso significativo entre o suposto começo do slide e o momento em que ele efetivamente está 100 por cento no ecrã, é, caso haja algum erro nos dados de um dos transístores poder-se sempre utilizar os dados do outro, não desperdiçando os dados da experiência em questão. Isto é feito verificando onde ocorre uma transição de 0 volts para um valor superior a 0 volts e quando ocorre a transição contrária - basicamente, quando ocorre uma transição de preto para branco e de branco para preto. Após isto verifica-se a distância entre transições, de forma a que apenas se considerem transições que transacionam novamente 2.4 ou mais segundos depois. A duração de cada slide referente a uma imagem cotada está programada para 2.43 segundos. Cria-se então uma matriz com os instante em que começam slides com uma imagem do [IAPS](#), ou seja, os instantes em torno dos quais se irá criar os segmentos.

Esta matriz é guardada com o nome kPosBegin. Há que ter em consideração que esta matriz contém o início dos slides da experiência inteira, onde é apresentado o slide-show 2 vezes - uma com o ecrã tapado e uma onde o participante vê o slide-show. Há que realizar o processamento em separado. Para tal, basta cortar a primeira ou a segunda metade da matriz kPosBegin antes de a guardar.

5.6.4 Faz02.5 - Filtra Canais

Este modulo permite isolar gamas específicas de frequência de forma a no final observar como essa seleção afeta os dados obtidos.

A filtragem é efetuada sobre os 40 canais iniciais. O processo está dividido em três partes.

Na primeira parte os ficheiro mSchs é separado 40 ficheiros - um por canal. Isto é apenas feito neste módulo e apenas este módulo utiliza estes ficheiros. Isto deve-se a, em testes iniciais, a matriz mSchs ser grande demais, o que atrasava demasiado a filtragem dos canais. Este efeito do tamanho de mSchs não se faz sentir tanto quando os canais estão a ser combinados, possivelmente por serem operações menos complexas.

Na segunda parte é criado o filtro. São especificadas as matrizes kFreqs e kAmps - frequências a considerar e respetiva amplitude. São devolvidos os coeficientes do filtro com as especificações desejadas. O filtro é guardado na pasta filter com o nome kFilter. Na terceira e ultima etapa de filtragem, o filtro criado é aplicado sobre os dados. Isto é feito aplicando convolução dos coeficientes do filtro sobre o sinal de cada canal. No entanto, a convolução estende o sinal, acrescentando amostras antes e depois do sinal original. Como tal o resultado da convolução é reduzido ao tamanho original do sinal. A característica de magnitude e e fase de todos os filtros utilizados pode ser observada no anexo E. Os filtros pretendem cortar diferentes gamas de frequência. O objetivo é verificar a influência que diferentes gamas de frequências poderão ter nos resultados finais. A escolha da característica do filtros não está baseada em nenhum critério em específico.

5.6.5 Faz03 - Combina Canais

Este módulo é responsável por calcular todas as combinações possíveis entre 40 valores diferentes retirados 2 a 2. Isto configurará a matriz de combinações. Após isto, cada combinação representará um canal bipolar - diferença entre os dois canais.

Como foi referido na secção referente à organização, cada canal é guardado num ficheiro .mat separado. Os primeiros 40 canais são mono polares e os 780 seguintes são bipolares.

5.6.6 Faz04 - Segmenta Canais

Agora que já se tem todos os canais mono polares e bipolares, pode-se extrair apenas os bocados de cada canal que são relevantes - os intervalos em torno dos instantes extraídos em Faz02.

A pasta dos segmentos mantém-se sempre igual, sendo apenas substituídos os ficheiros pelos ficheiros relativos ao take a decorrer. Neste modulo é feito o load da lista de takes - kTakeList - e são extraídos os valores que deverão ser utilizados como “qWindow” e “qBackWindow”.

Os segmentos irão preencher uma matriz $71 \times T_{\text{seg}}$ uma vez que temos 71 slides.

Pode-se também definir $T_{\text{seg}} = nWindow + nBackWindow + 1$, uma vez que o instante em que começa o slide deverá ser considerado no tamanho do segmento.

Respeitando a semântica de Matlab®, pode-se definir o segmento relativo ao slide i como sendo - $s_i = V_{\text{exp}}(ki32PosBegin(i) - nBackWindow : ki32PosBegin(i) + nWindow)$.

5.6.7 Faz05 - Calcula Estimadores

Este módulo está responsável por construir o primeiro esboço do que serão os estimadores a utilizar. De forma a evitar obter só acertos em cada teste que se realize, deve-se construir um estimador para cada slide, que não contenha o próprio. De forma a realizar isto é suposto subtrair o segmento a avaliar do estimador que lhe diz respeito. Isto é efetuado no módulo seguinte.

O cálculo do esboço dos estimadores - `mEstimateFull` - é realizado para cada nível, por cada canal. São percorridos todos os segmentos e são somados todos os que forem referentes ao mesmo nível.

Na pasta `estimateFull` são guardados todos os estimadores relativos a cada canal - 8 estimadores por canal.

5.6.8 Faz05.5 - Subtrai o Próprio

Neste módulo são criados um conjunto de estimadores por cada slide.

Cada slide terá então um conjunto de 8 estimadores (por cada nível). Cada segmento está contido num estimador - está contido no estimador do nível associado à imagem [IAPS](#) em questão. Portanto, cada segmento deverá ser subtraído do estimador (criado no módulo anterior) que lhe diz respeito. Os estimadores dos outros níveis são copiados diretamente dos estimadores em bruto, provenientes do módulo anterior - `mEstimateFull`.

Os estimadores específicos de cada segmento - `mEstimateSpecific` - terão que ser adaptados de forma a terem média nula e estarem normalizados à energia unitária.

Caso o segmento seja referente a uma imagem sem nível associado - uma imagem que não seja do [IAPS](#) - então não é subtraído o segmento, uma vez que este não está contido em nenhum estimador. Para estes slides são considerados os estimadores `mEstimatesFull`, com média nula e energia unitária.

Os estimadores específicos são guardados da seguinte forma - uma pasta para cada canal, em que, em cada pasta estão 71 conjuntos de 8 estimadores, ou seja, 8 estimadores por slide.

5.6.9 Faz06 - Aplica Estimadores

Por fim, é necessário aplicar os estimadores aos segmentos. O objetivo será no fim ter um vetor de decisões que o pipeline tomou. Estas decisões são o nível que o pipeline atribuiu a cada segmento. É criado um vetor de decisão - `kResults` - por cada canal.

Esta decisão é tomada da seguinte forma:

É criado um vetor de votos. Cada voto é o resultado do produto interno entre cada um dos estimadores e o segmento respetivo. Por cada canal são carregados todos os segmentos e, por cada segmento, são carregados todos os estimadores.

Deste vetor de votos é considerado o índice do maior voto, ou seja, é considerado o nível do estimador que tem maior correlação com o segmento.

Este processo é repetido para cada canal. Os resultados são guardados na mesma pasta; isto equivale a uma vetor de 71 decisões por cada canal.

5.7 Elementos Variáveis

O pipeline pode correr um determinado número de vezes. Por cada vez são considerados valores diferentes para certos elementos variáveis. São chamados recursivamente os módulos em que estes valores terão influência. Os elementos que são selecionáveis por quem corre o pipeline são:

- Janelas - Como foi dito, é um conjunto de pontos no tempo onde os slides começaram a ser exibidos. O utilizador poderá considerar diferentes intervalos relativos a cada um destes pontos. Deverá ser formado no início do pipeline um vetor que contenha um conjunto de pontos com dois valores: quantas amostras considerar antes - qBackWindow - do ponto de início de cada slide e quantas amostras considerar depois - qWindow. O valor de qBackWindow poderá ser negativo, significando isto que só se consideram valores após o início de cada slide. Considerar um intervalo novo de segmentação pressupõe correr repetidamente os módulos: Faz04, Faz05, Faz05.5 e Faz06.
- Filtro - O utilizador poderá também selecionar um conjunto de valores diferentes para a frequência de corte do filtro passa-baixo e do filtro passa-alto. Isto permite selecionar o intervalo de frequências que irão passar por aquilo que acaba por ser um filtro passa-banda : passa-baixo mais passa-alto. Os valores a escolher são, para o corte a altas frequências e baixas frequências respetivamente: qHighCut e qLowCut. À semelhança das janelas, deverá ser criado um vetor com diversos pontos num espaço de duas dimensões possíveis. Considerar um novo intervalo de frequências pressupõe correr repetidamente os módulos: Faz01.5, Faz03 Faz04, Faz05, Faz05.5 e Faz06.

Os resultados que provêm dos diferentes elementos variáveis são guardados em pastas diferentes, identificadas com os valores destas variáveis. É criada uma pasta por cada intervalo de frequência, na qual são criadas diferentes pastas por cada intervalo de amostras considerado

5.8 Software de Teste

O software de teste irá testar o desempenho do software em situações artificiais. Para tal, os valores lidos pelos elétrodos deverão (, caso já tenham sido realizadas leituras de ficheiros .bdf) ser substituídos por sinais gerados.

Para cada canal, o sinal lido deverá ser substituído por um sinal de ruído gaussiano. É criado um módulo adicional, cujo objetivo é substituir o sinais lidos de cada um dos

40 canais. A ideia chave é criar um sinal representativo de cada nível. Estes sinais são somados aos canais nos momentos em que slides com o nível respectivo tenham sido mostrados.

O interesse está em criar situações em que os sinais devam ser todos corretamente classificados (valor alto de **SNR** - Signal to Noise Ratio) e situações em que apenas o acaso é responsável pelas classificações corretas (valor baixo de **SNR**) - isto permite validar o software como estado a funcionar de acordo com a teoria.

No entanto, o interessante está também em utilizar este software para verificar em que valor de **SNR** é que se deixa de recuperar a mensagem na íntegra.

5.8.1 Fazxx - Auxiliar de Teste

Este módulo corre entre os módulos Faz02 e Faz03. É carregada a matriz mSchs. Os 40 canais são substituídos por ruído branco gaussiano.

Após isto, é criado um sinal aleatório para cada nível de agrado e para cada canal. Estes sinais deverão ter tamanho equivalente à menor duração registada de uma imagem **IAPS**. Na matriz mSchs, a cada intervalo correspondente a uma imagem **IAPS** deverá ser adicionado o sinal do nível respectivo. O ruído que corrompe o sinal de nível é o ruído inalterado presente na matriz mSchs. Este ruído não será alterado. É mais prático alterar o sinal de nível antes de o somar ao ruído já presente em cada canal.

Recorrendo à definição de rácio sinal/ruído apresentada em 3.5 no Capítulo 3, serão demonstradas de seguida as operações a realizar sobre cada canal.

De forma a obter um sinal que tenha potência **SNR** vezes superior ao ruído:

$$S_{sinal} = SNR \times S_{ruído}. \quad (5.1)$$

Relembra-se a definição de energia - $E = \sum_{n=-\infty}^{\infty} x[n]^2$. Visto que o ruído e o sinal têm a mesma duração, a equação 5.1 será verdadeira se se multiplicar a potência do sinal de nível por um fator k:

$$k = SNR \times \frac{E_{ruído}}{E_{sinal}}. \quad (5.2)$$

No entanto, apenas se pode aceder e manipular os valores temporais do sinal de nível, portanto, a equação 5.1 será verdadeira se:

$$S_{sinal} = \frac{1}{N} \sum_{n=1}^N (\sqrt{k} x_{sinal}[n])^2. \quad (5.3)$$

Portanto, apenas se tem que descobrir o fator k para cada segmento e multiplicar o sinal de nível pela raiz quadrada de k.

De notar também que a relação sinal ruído vem em dB pelo que ainda é necessário converter no respetivo valor de ganho $G = 10^{\frac{qSNR}{20}}$. A relação sinal ruído entre o sinal do nível e o ruído branco é um elemento variável exclusivo do software de teste. Isto permite

que sejam realizados testes diferentes para diferentes valores de sinal ruído. Desta forma pode-se verificar em que valor da relação sinal ruído o software deixa de ser capaz de reconstruir a mensagem. O utilizador passa a poder escolher diferentes valores para a relação entre o sinal de nível e o ruído branco. A matriz de valores de SNR - kSNR - é guardada no início do correr do pipeline, assim como o número de testes a realizar para diferentes valores - qSNRTakes. Deverá ser guardado também o valor relativo ao rácio atual - qCurrentSNR.

Para cada valor diferente de SNR serão corridos os módulos Fazxx, Faz03 Faz04, Faz05, Faz05.5 e Faz06.

ANALISE DE RESULTADOS

Este capítulo está dedicado à apresentação dos resultados obtidos.

6.1 M-sequência gerada

As três M-sequência geradas e a mensagem correspondente podem ser observados no Anexo F. É importante reparar que, uma vez que a mensagem tem 63 símbolos de comprimento, todos os níveis terão 8 imagens no slide-show, exceto o nível 9 que só tem 7. Como seria teoricamente esperado, verifica-se pela figura 6.1 que a autocorrelação circular das três m-sequências aproxima um impulso. Correspondendo à autocorrelação circular relativa a uma sequência binária pseudo-aleatória.

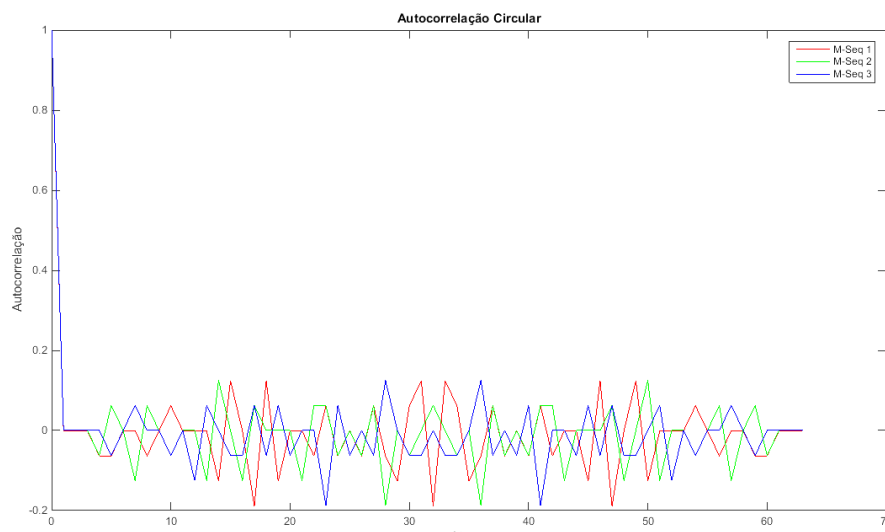


Figura 6.1: Autocorrelação circular das 3 m-sequências geradas

Máximos			
Auto correlação	1	0.1887	0.1887
Amostra	1	18	48
Mínimos			
Auto correlação	0.0010	0.0010	0.0010
Amostra	62	63	64

Tabela 6.1: Picos e mínimos na autocorrelação circular da sequência 1

Máximos			
Auto correlação	1	0.1887	0.1887
Amostra	1	29	37
Mínimos			
Auto correlação	0.0010	0.0010	0.0010
Amostra	19	47	64

Tabela 6.2: Picos e mínimos na autocorrelação circular da sequência 2

Máximos			
Auto correlação	1	0.1875	0.1875
Amostra	1	24	42
Mínimos			
Auto correlação	0	0	0
Amostra	62	63	64

Tabela 6.3: Picos e mínimos na autocorrelação circular da sequência 3

Verifica-se alguns picos na auto correlação, dependendo do atraso. Todos estes picos são inferiores a 0.2, como se pode observar pelas tabelas 6.1, 6.2 e 6.3. Nas tabelas também está indicada a amostra em que se detetou os máximos e mínimos da autocorrelação correspondente correspondentes.

A auto correlação da sequência criada a partir das três m-sequências geradas é apresentada na figura 6.2:

É possível verificar que a auto correlação da sequência final aproxima um impulso, atingindo o valor máximo no instante inicial e valores menores em todos os outros instantes. A auto correlação apresenta, no entanto, alguns picos maiores que as sequências binárias. Isto deve-se ao facto de o método utilizado ser uma aproximação a uma sequência aleatória de 8 símbolos diferentes. Mas para o simples objetivo de aleatorizar a experiência - como tinha sido referido no capítulo 2 - é uma aproximação suficiente.

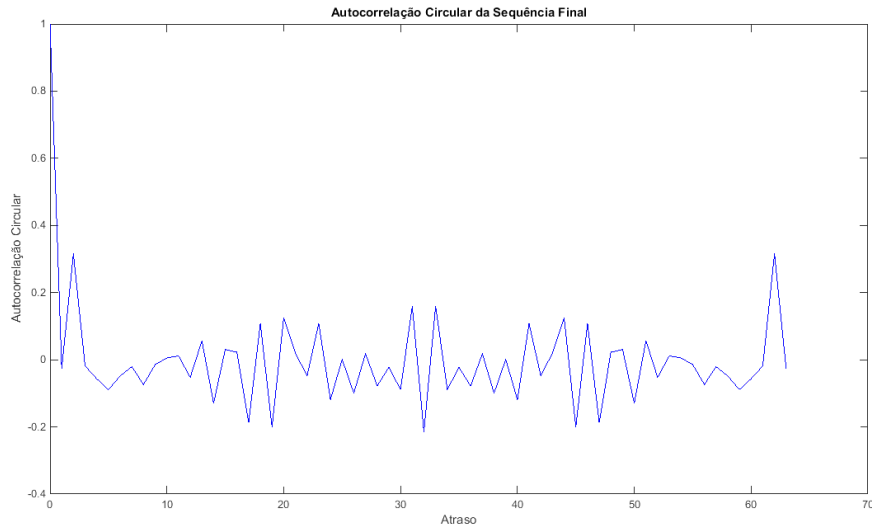


Figura 6.2: Autocorrelação circular da sequência gerada a partir das 3 m-sequências

Máximos			
Auto correlação	1	0.3154	0.3154
Amostra	1	3	63
Mínimos			
Auto correlação	0.0052	0.0009	0.0009
Amostra	55	26	40

Tabela 6.4: Picos e minimos na autocorrelação circular da sequência final

6.2 Considerações Globais

Devido a dados defeituosos proveniente do fototransistor do canto inferior direito (fim da mudança de slide), apenas foi utilizado o fototransistor do canto superior esquerdo (inicio da mudança de slide).

De forma a minimizar o número de tabelas neste capítulo, foram apenas apresentados os resultados de sujeitos experimentais relativos a uma filtragem de 4 a 50 Hz e aos intervalos - [300 400], [400 500] e [300 500] milissegundos.

Este preciso set de experiências é relativo à gama de frequências - 4 a 50 Hz - onde se inserem grande parte dos diferentes tipos de ondas cerebrais [13].

Como foi dito, não é o objetivo detetar o P300. No entanto, os resultados que têm maior relevo segundo a literatura apresentada são do set de intervalos de tempo apresentado.

Outros resultados foram obtidos e podem ser observados no anexo I. Também é possível observar a ocorrência dos slides durante as três experiências no anexo H. As experiências foram realizadas sobre três participantes experimentais, masculinos, estudantes, de idades compreendidas entre os 24 e 28 anos.

6.3 Validação do Software

De forma a validar o software foram realizados testes, prévios à sua utilização em participantes.

Estes testes têm por intuito validar que o software funciona em situações óbvias (i.e. situações em que é esperado que a mensagem seja recuperada na totalidade). Têm também o intuito de mostrar que o software não recupera a mensagem em situações em que não há relação entre a mensagem e o sinal lido.

6.3.1 Testes com sinais de nível gerados

Os sinais foram gerados recorrendo ao módulo auxiliar de teste descrito no capítulo 5.

Estes testes, caso se comportem como teoricamente esperado, permitem validar o software enquanto método de deteção e classificação.

O software de teste permite criar dados que serão associados aos diferentes níveis IAPS. Estes sinais serão corrompidos com ruído gaussiano branco. Será de esperar que para valores altos de SNR o sinal seja detetado com facilidade, e, à medida que o SNR decresce, também deverá decrescer o número de acertos, até ao mínimo de aproximadamente 8. Para estes testes não foi considerada nenhuma filtragem, uma vez que não é relevante isolar nenhum conjunto de frequências quando se está a trabalhar sobre sinais gerados.

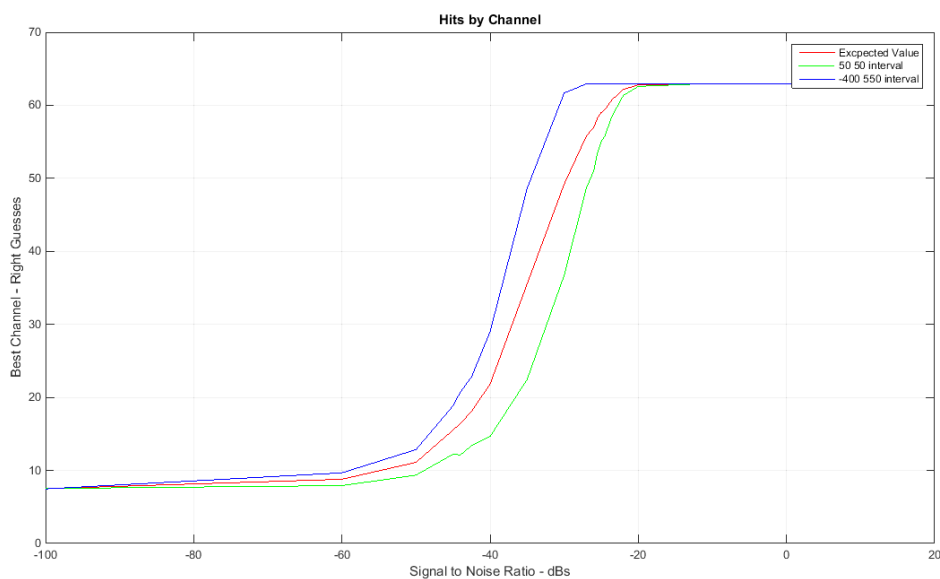


Figura 6.3: Valor médio de assertos de todos os canais. Gráficos para cada um dos dois intervalos diferentes e para a média entre os resultados de cada um.

SNR	-100	-60	-48	-44	-42.5	-40	-35	-30	-27	-26	-25.5	-24.5
Acertos	18.5	20.5	29.0	31.0	32.5	36.0	47.0	58.0	61.0	61.5	62.0	63.0

Tabela 6.5: Número de pontos mínimos representativos da recta de valor esperado dos dois intervalos 6.3

A figura 6.3 permite observar o que seria teoricamente esperado - O número de acertos é diretamente proporcional ao valor de SNR. É importante verificar a diferença entre os resultados do intervalo de [50 50] ¹ em relação ao intervalo [-400 550] ². Os resultados são significativamente mais baixos para o intervalo de [50 50]. Isto deve-se ao facto de metade do intervalo [50 50] ser apenas ruído sem sinal gerado. Já o intervalo [-400 550] não tem nenhum momento em que seja apenas ruído.

Isto permite compreender a sensibilidade do método à escolha de intervalos. Será então de extrema importância sincronizar o método, não só com o início dos slides (como já é considerado), mas também com o momento em que surge a manifestação de agrado.

É apresentada uma possível solução para este problema no capítulo 7.

Também é importante verificar que, para a criação dos estimadores, os mesmos sinais são somados 8 vezes (para todos os estimadores, exceto o de nível 9). Isto pelos resultados obtidos na equação 3.10 significa que o pipeline de processamento funciona para casos em que o rácio sinal ruído seja 8 vezes inferior ao apresentado na figura 6.3. Segundo este raciocínio, onde se verifica que, para o gráfico do valor esperado, o pipeline deixa de conseguir recuperar a mensagem completa para -25.5dB entre o sinal de nível e o ruído do segmento (aproximadamente 0.053 em decimal), na verdade este valor corresponde a $0.053 \times 8 = 0.424$ de SNR entre o ruído do estimador e o sinal de nível, ou seja, aproximadamente -7.45 dB.

Os resultados obtidos para a classificação em condições de SNR variável são concordantes com os resultados obtidos para a emulação de um canal em que os sinais a detetar são corrompidos por ruído branco [4].

A figura 6.4 serve para o leitor observar a expectável semelhança entre os estimadores e os sinais referentes ao nível em questão. Esta semelhança é logicamente mais visível para valores altos de SNR.

Verifica-se, pela tabela 6.6, que, com o aumento da potência do ruído, relativamente à potência do sinal, a correlação entre os estimadores e o sinal de nível vai reduzindo. Este fenómeno é justificado, uma vez que um menor SNR implica, logicamente, que os estimadores sejam constituídos por mais ruído que sinal. O ruído introduz componentes não correlacionadas, pelo que, quanto mais ruído houver, menor será a correlação entre o estimador e o sinal puro (i.e. sinal de nível).

¹ 50 milissegundos antes do sinal gerado até 50 milissegundos depois

² 400 milissegundos após o início do sinal gerado até 550 milissegundos após o início do sinal gerado

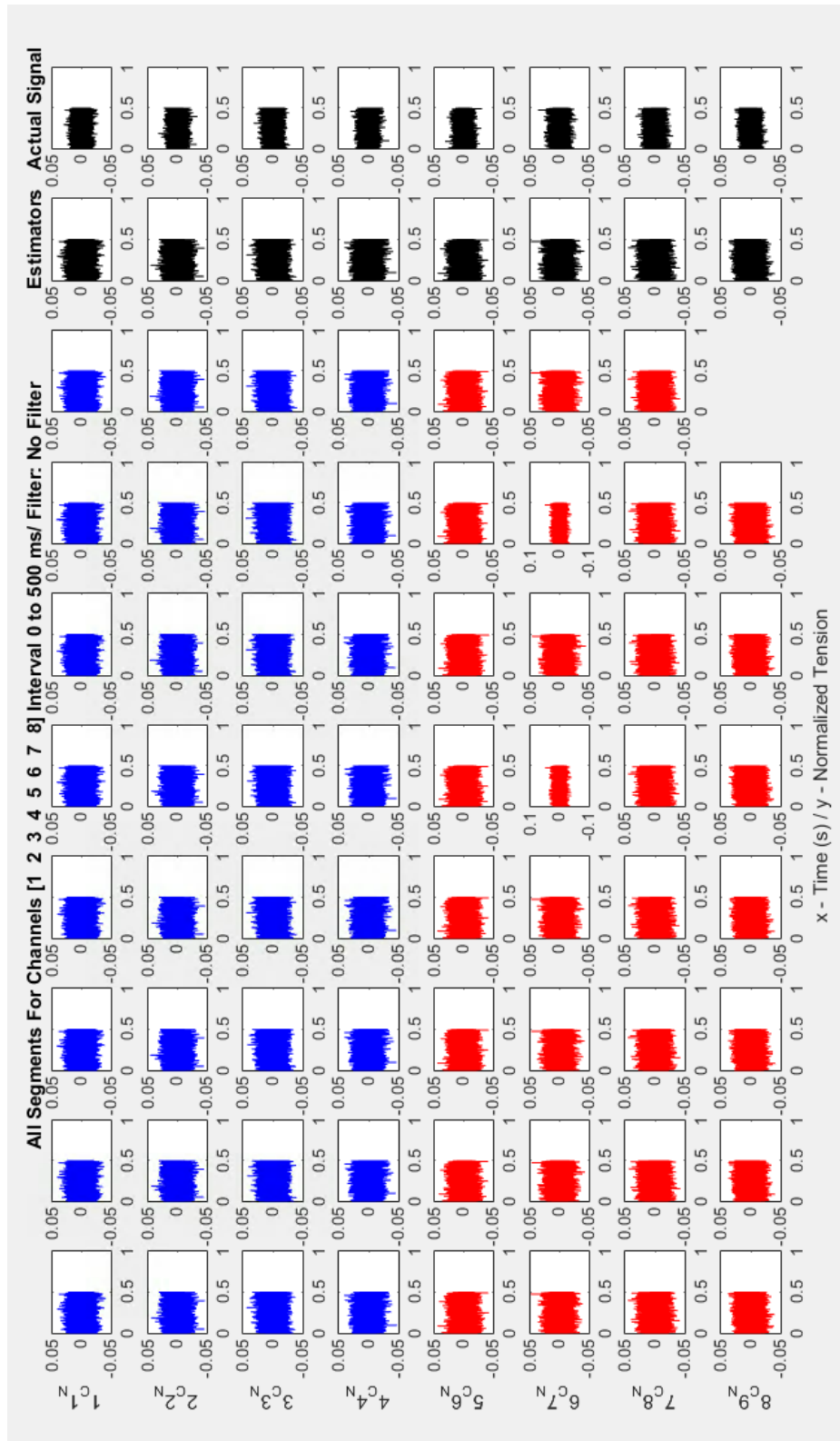


Figura 6.4: Figura puramente representativa da semelhança entre os estimadores e os sinais gerados quando o SNR é alto - 40dB para o caso da figura. A vermelho e azul - os segmentos que dão origem aos estimadores. A preto - os estimadores e os sinais de nível. Cada linha de gráficos é um canal/nível diferente.

SNR \ Nível	1	2	3	4	6	7	8	9
40	0.729	0.734	0.727	0.732	0.722	0.733	0.731	0.725
20	0.723	0.716	0.724	0.719	0.716	0.729	0.721	0.721
0	0.687	0.688	0.687	0.686	0.682	0.681	0.676	0.675
-20	0.482	0.484	0.486	0.480	0.493	0.497	0.481	0.476
-60	0.070	0.057	0.066	0.074	0.067	0.062	0.071	0.072

Tabela 6.6: Correlação entre os estimadores e os sinais de nível gerados para os canais [1 2 3 4 5 6 7 8]

6.3.2 Teste em Fontes de Correlação Aproximadamente Nula Expectável

Foram realizados testes sobre fontes, cujo sinal gerado/lido não deveria conter nenhuma manifestação das imagens apresentadas.

Estes testes pretendiam mostrar que o pipeline utilizado obteria um resultado puramente estocástico - nesta situação (i.e. 63 slides, com 8 opções de escolha possíveis) é expectável que se decida corretamente sobre 8 slides.

6.3.2.1 Teste com Bucket

O teste sobre o chamado Single Bucket diz respeito à realização da experiência com todos os elétrodos submersos em água salgada, dentro de um balde. O slide-show foi apresentado, de forma que se tenha todos os dados necessários para correr o pipeline.

Isto permite verificar eventuais falhas no equipamento (falhas internas, falhas nos elétrodos e falhas nos conectores) [10].

Sem Filtro	Moda	Média	Máximo	Canal
take-500to900	2	6.12	17	6 e 27
take0to1200	0	4.46	15	3 e 30
take-1000to2600	0	4.05	14	29 e 30
take400to2600	0	3.09	12	1 e 18
take0to50	4	6.96	17	2 e 25
take-300to550	4	7.37	18	6
take-200to1000	0	5.29	16	2

Tabela 6.7: Resultados para o Single Bucket

A tabela 6.7 apresenta o número de acertos obtidos considerando a moda da votação de todos os canais (i.e. votação conjunta de todos os canais), a média dos acertos obtidos de todos os canais, o máximo de acertos obtidos entre todos os canais e o respetivo canal. No entanto, neste caso, o objetivo principal deste teste é verificar se o resultado corresponde ao resultado esperado para uma leitura puramente ruidosa.

Será importante relevar na tabela 6.7 a diferença entre os resultados, dependendo do quão grandes são os intervalos - os resultados da moda das votações, a média de

acertos e os acertos máximos são menores quanto maior for o intervalo considerado. Seria esperado que a média dos acertos fosse aproximadamente 8. Também seria de esperar que o máximo estivesse entre o intervalo $\{18, 22\}$. Este fenómeno pode indicar uma de duas coisas: o método que se está a considerar é sensível ao tamanho dos segmentos, ou, o hardware está a introduzir algum tipo de interferência nas leituras realizadas.

Resultados adicionais são apresentados no anexo G, nos quais se pode verificar que quando é aplicada filtragem sobre os dados, os resultados se aproximam da situação esperada (i.e. votação aleatória).

6.3.2.2 Teste Sobre Sinais Gerados Aleatórios

Esta secção está destinada a apresentar resultados de testes em que o pipeline foi aplicado a ruído. Em termos de implementação isto implica apenas trocar os dados provenientes de uma experiência por ruído branco gaussiano.

Será teoricamente esperado que se obtenha resultados máximos dentro do intervalo $\{18, 22\}$ e médios de aproximadamente 8 acertos, uma vez que não existe qualquer relação entre os níveis e o ruído.

	Moda	Média	Máximo	Canal
take-500to900	11	8.43	20	8 e 09
take0to1200	8	7.86	20	4 e 20
take-1000to2600	14	8.45	23	14 e 25
take400to2600	5	7.88	18	4 e 06
take0to50	11	8.17	21	24 e 29
take-300to550	7	7.90	20	2 e 18
take-200to1000	9	8.50	21	4 e 00

Tabela 6.8: Resultados para testes sobre ruído branco

Os resultados 6.8 encontram-se dentro do teoricamente esperado. Os testes foram realizados para os mesmos takes que o Single Bucket de forma a que os resultados sejam comparáveis.

6.4 Resultados em Participantes Experimentais

De seguida são apresentados os resultados obtidos aplicando o software criado sobre os dados dos participantes experimentais.

Cada experiência diferente é chamada de "EXP", seguida de um número que seja representativo da ordem cronológica em que a experiência foi realizada. Isto permite preservar o anonimato de quem realizou a experiência. Permite também ter uma noção relativa de que experiências foram executadas primeiro.

4 - 50 Hz	Moda	Média	Máximo	Canal
take-300to400	16	10.20	22	31 e 37
take-400to500	5	7.30	19	9
take-300to500	17	9.90	20	1 e 19

Tabela 6.9: Resultados para a EXP01

São apresentados 3 resultados fundamentais por experiência:

- Tabela com os resultados para 3 takes diferentes - filtragem de 4-50 Hz;
- Gráficos dos melhores canais para cada nível e o respectivo estimador resultante;
- Tabela com a probabilidade de erro de cada estimador apresentado.

O objetivo dos gráficos da figura 6.5 é poder apresentar um caso exemplificativo, mas também selecionar os melhores casos. Pode-se inferir que os melhores resultados virão dos melhores estimadores. Foi selecionado o 3º take - [300 500] - para apresentação por obter, regra geral, os melhores resultados. A figura (e todas as seguintes 6.6 e 6.7) foram criadas da seguinte forma: O pipeline é corrido e são obtidos resultados para cada filtragem e take. Com auxílio das funções criadas, é calculada a taxa de erros por nível para cada canal. É guardado o melhor nível de cada canal, assim como o número de acertos para o respectivo nível.

No final deste processo é possível escolher um canal por nível. O critério passa por escolher o canal que tenha obtido um maior número de acertos para o nível em questão. Isto permite selecionar 8 canais (um por nível) e apresentar todos os segmentos que deram origem ao respectivo melhor estimador.

São também apresentados os estimadores de cada nível na última coluna de forma a que se possa observar o resultado final dos segmentos apresentados. Isto permite ao utilizador observar a existência de padrões nos sinais para cada momento em que é apresentada uma imagem do nível em questão.

	P(Erro)	Acertos
1	0.250	6
2	0.000	8
3	0.250	6
4	0.250	6
6	0.375	5
7	0.500	4
8	0.250	6
9	0.429	4

Tabela 6.10: Probabilidade de Erro e Acertos Para cada Nivel - EXP01

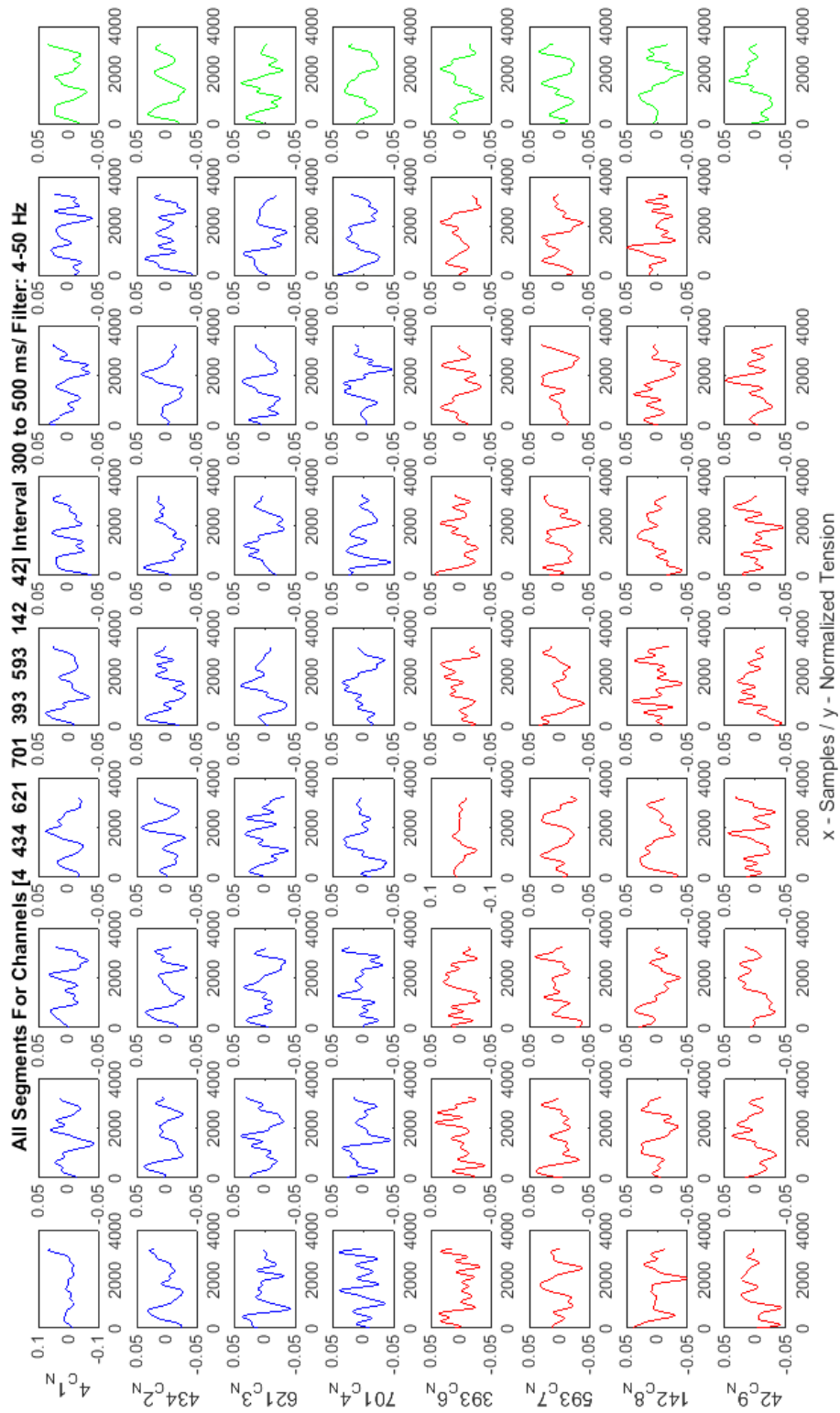


Figura 6.5: Segmentos e estimadores dos melhores níveis dos melhores canais - EXP01

Por fim é apresentada uma tabela de probabilidade de erros e acertos para cada um dos estimadores apresentados na figura 6.5.

Isto permite ter uma ideia da probabilidade de erro por nível - caso se utilize cada um dos estimadores de cada um dos canais.

4 - 50 Hz	Moda	Média	Máximo	Canal
take-300to400	11	7.71	18	19 e 26
take-400to500	7	8.21	19	15 e 33
take-300to500	13	8.43	21	4 e 26

Tabela 6.11: Resultados para a EXP02

Os resultados das tabelas Moda/Média/Máximo 6.9, 6.11 e 6.13 parecem todos ser relativos a votações aleatórias. É possível notar, no entanto, que os melhores resultados são obtidos na EXP01 e os piores para a EXP03.

Também é possível notar que para a experiência 6.11 o take que obteve melhores resultados é [300 500]. No entanto, para a experiência 6.9, o take [300 500] é o melhor para os resultados da moda e o take [300 400] é o melhor para os resultados do melhor canal e da média. O take [400 500] é responsável pelos piores resultados da experiência 01. O take [300 400] obtém os piores resultados para a média e máximo da EXP02.

	P(Erro)	Acertos
1	0.375	5
2	0.250	6
3	0.250	6
4	0.125	7
6	0.250	6
7	0.375	5
8	0.375	5
9	0.143	6

Tabela 6.12: Probabilidade de Erro e Acertos Para cada Nível - EXP02

4 - 50 Hz	Moda	Média	Máximo	Canal
take-300to400	2	7.26	18	24 e 38
take-400to500	5	7.21	16	23 e 27
take-300to500	4	6.81	17	22 e 27

Tabela 6.13: Resultados para a EXP03

A EXP03 parece obter resultados evidentemente inferiores aos resultados das outras duas experiências, tanto a nível de moda, como de média e máximo.

Poderia-se dizer que é a experiência que mais se aproxima de uma situação aleatória - Ver

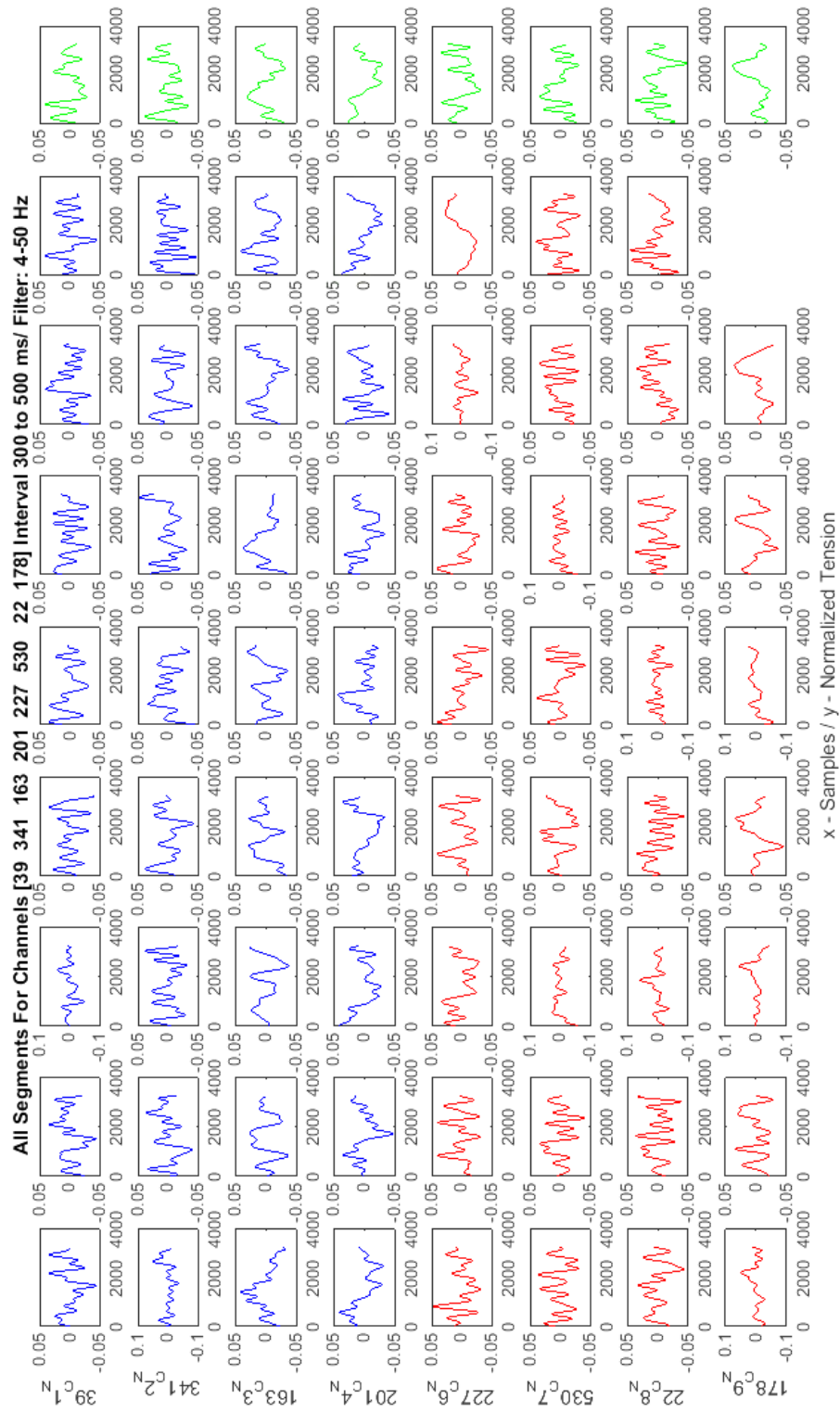


Figura 6.6: Segmentos e estimadores dos melhores níveis dos melhores canais - EXP02

resultados para sinais aleatórios.

Como tal, é possível observar que a soma das probabilidades de erro é superior à das duas outras experiências.

	P(Erro)	Acertos
1	0.375	5
2	0.125	7
3	0.500	4
4	0.375	5
6	0.375	5
7	0.375	5
8	0.250	6
9	0.286	5

Tabela 6.14: Probabilidade de Erro e Acertos Para cada Nivel - EXP03

6.5 Discussão de Resultados

As sequências criadas aproximam o caso aleatório que era pretendido. Não será necessário muito rigor para o aleatório obtido, uma vez que apenas se quer evitar que o participante tenha noção de possíveis padrões na apresentação das imagens, o que poderia causar uma resposta prévia à imagem seguinte. Isto tornaria o sistema (i.e. participante experimental) não causal.

Os resultados do Single Bucket revelaram que, quando apenas é lido ruído dos elétrodos, o pipeline obtém resultados ainda piores que a situação aleatória esperada. Isto pode estar relacionado com possíveis artefactos introduzidos pelo sistema de aquisição, pelo que esta possibilidade deve ser verificada de seguida, como será sugerido no capítulo 7. É também relevante referir a importância de analisar o hardware em questão - uma diferença entre o equipamento BioSemi utilizado e outros sistemas de EEG é a utilização de dois elétrodos de referência em vez do clássico eléctrodo de "ground". Os dois elétrodos de referencia consistem em um elemento ativo - CMS (common mode sense)- e um elemento passivo - DLR (driven right leg) [3]. Isto trás algumas vantagens aos métodos de referencia clássicos. No entanto, há que verificar a influência que o eléctrodo ativo possa ter sobre os dados lidos.

É possível verificar a clara diferença entre os resultados dos testes realizados com os elétrodos em água salgada e dos testes realizados sobre ruído branco gaussiano gerado. Para os mesmos intervalos os resultados do valor máximo e média de acertos do Single Bucket são consideravelmente mais baixos. Também é possível verificar que, nos testes sobre sinais aleatórios gerados, não existe qualquer relação visível entre o tamanho dos intervalos e os resultados, ao contrário dos resultados do Single Bucket.

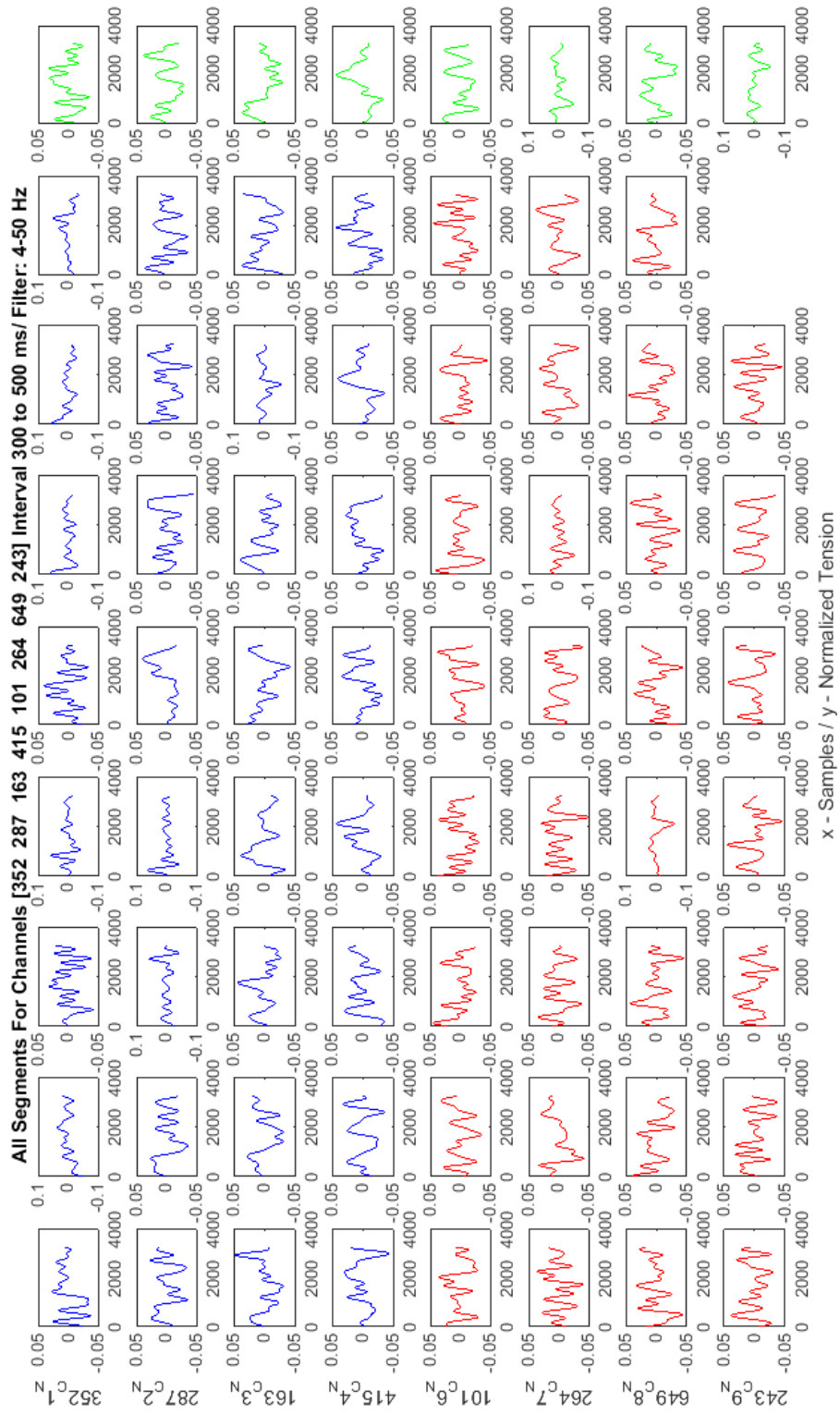


Figura 6.7: Segmentos e estimadores dos melhores níveis dos melhores canais - EXP03

Quanto aos resultados dos participantes experimentais: não parece ser possível extrair nenhuma conclusão dos resultados obtidos. Apesar de se obter resultados superiores aos resultados esperados para situações aleatórias - em particular na EXP01 -, não parece ser possível extrair conclusões sobre os mesmos. É assumida como verdadeira a hipótese nula apresentada no capítulo 1, uma vez que os resultados não permitem encontrar relação entre os sinais lidos e os níveis IAPS.

Apesar de ser possível construir um sistema com os melhores estimadores dos melhores canais 6.5 6.6 6.7, os resultados em si não parecem ser significativos ao ponto de não se considerar que os melhores resultados foram obtidos por pura sorte.

Os resultados sobre os dados relativos à primeira parte da experiência - em que o participante não vê os slides - correspondem ao caso aleatório, como se pode verificar pelo anexo J.

Os dados da moda foram apresentados por serem relativos a uma possível forma de votação em que cada canal “vota” sobre o nível de cada segmento. Este valor não tem nenhum significado particular para os resultados apresentados. Não é possível definir um valor esperado para o resultado da moda, como se pôde verificar pela variância dos mesmos.

O trabalho futuro deve estar focado em resolver e averiguar problemas no hardware, antes de todos os outros possíveis problemas.

Os possíveis problemas do método/pipeline/modelo utilizado assim como o trabalho sugerido para cada caso são apresentados no capítulo 7.

CONCLUSÃO

7.1 Contribuições

Quanto às contribuições consideradas no capítulo 1, é possível confirmar que se tem um sistema de processamento de dados válido para o modelo considerado. Esta conclusão é fundamentada pelos resultados obtidos em 6.3 e em 6.8. Estes dois testes correspondem ao conjunto de validação do método para o modelo apresentado. Isto, uma vez que, para o caso de sinais corrompidos em ruído branco, a fundamentação teórica 3 mostra que a média do sinal é o melhor estimador. Quanto à correlação entre o estimador e os segmentos é de esperar que quanto maior for o ruído em relação ao sinal, menor será a correlação entre o estimador e os segmentos, uma vez que o ruído introduz componentes não correlacionadas, como está comprovado pela tabela 6.6. Quando o pipeline foi executado sobre dados aleatórios gerados, obteve-se o que era teoricamente esperado - não há correlação considerável entre os estimadores e os segmentos, pelo que, para 8 valores diferentes e um grande conjunto de testes realizados deve obter-se aproximadamente 8 resultados corretos em 63 possíveis. Isto é confirmado pelos resultados 6.8.

A metodologia apresentada pode ser aproveitada por conter um conjunto de boas práticas, derivadas do conhecimento colecionado das áreas da engenharia, psicologia e neurociência. Esta dissertação pode, até certo ponto, ser utilizada como guia experimental para experiências que utilizem EEG. Pode também ser utilizada como guia de boas práticas tanto para a construção de um pipeline de processamento de dados, mas também para o processamento dos dados.

No entanto, esta dissertação falha em criar um método fiável de classificação de níveis de agrado. As possíveis soluções são apresentadas em 7.2, pelo que se reforça a necessidade da análise ao hardware estar no topo da lista de prioridades para trabalho futuro.

7.2 Trabalho Futuro

A investigação sobre este método ainda está numa fase demasiado recente para que se exclua a hipótese não nula apresentada no capítulo 1.

De seguida são apresentados alguns fatores que podem ser modificados de forma a verificar se o modelo consegue obter um conjunto de decisões mais exatas:

- Verificar se o modelo de ruído aditivo é o mais adequado - Isto implicará repensar todo o processo (i.e. redução do ruído, através da média dos sinais e a criação dos estimadores que só fazem sentido caso o ruído seja aditivo);
- Verificar se o ruído é gaussiano branco - o ruído pode assumir diversas distribuições, não necessariamente gaussianas. Isto pode ser verificado testando diferentes modelos, e, consequentemente, diferentes métodos de estimação;
- Verificar se o hardware não está a introduzir artefactos - a preocupação por este problema é sustentada pelos resultados 6.7. A verificação deste problema deve passar por realizar os testes adicionais que se encontram em [10]. Deve também ser complementada com investigação adicional sobre o sistema de aquisição de dados em questão. É necessária investigação suficiente para que seja possível ter alguma expectativa sobre o comportamento dos elétrodos;
- Verificar os “timings” do início dos segmentos - mesmo que não se procure o P300 nos sinais lidos para classificação, este sinal, caso seja captado por algum dos elétrodos, pode ser utilizado como um indicativo do processamento cerebral de um estímulo recebido. Como tal os segmentos e, consecutivamente, os estimadores podem ser calculados desde a deteção do P300 até um comprimento definido.

Todas estas possíveis soluções implicam testar diferentes hipóteses. É do meu intuito incentivar a continuação de investigação nesta área.

O facto de se poder encontrar muitos resultados positivos na investigação realizada na área da classificação de estímulos incide algumas luzes na possibilidade de criar um método que atinja resultados satisfatórios.

BIBLIOGRAFIA

- [1] R. L. Ackoff. "From data to wisdom". Em: *Journal of applied systems analysis* 16.1 (1989), pp. 3–9.
- [2] I. Bakker, T. van der Voordt, P. Vink e J. de Boon. *Pleasure, Arousal, Dominance: Mehrabian and Russell revisited*. 2014. DOI: [10.1007/s12144-014-9219-4](https://doi.org/10.1007/s12144-014-9219-4).
- [3] Biosemi Home Page. <http://www.biosemi.com>.
- [4] A. Boscolo, F. Vatta, F. Armani, E. Viviani e D. Salvalaggio. "Physical AWGN Channel Emulator for Bit Error Rate Test of Digital Baseband Communication". Em: *Applied Mechanics and Materials* 241-244 (2012), pp. 2491–2495. ISSN: 1662-7482. DOI: [10.4028/www.scientific.net/AMM.241-244.2491](https://doi.org/10.4028/www.scientific.net/AMM.241-244.2491). URL: <http://www.scientific.net/AMM.241-244.2491>.
- [5] M Bradley e P. J. Lang. "Measuring Emotion: The Self-Assessment Semantic Differential Manikin and the". Em: *Journal of Behavior Therapy and Experimental Psychiatry* 25.1 (1994), pp. 49–59. ISSN: 00057916. DOI: [10.1016/0005-7916\(94\)90063-9](https://doi.org/10.1016/0005-7916(94)90063-9). arXiv: [0005-7916\(93\)E0016-Z](https://arxiv.org/abs/0005-7916(93)E0016-Z). URL: <http://www.sciencedirect.com/science/article/pii/0005791694900639>.
- [6] H. Candra, M. Yuwono, A. Handojoseno, R. Chai, S. Su e H. T. Nguyen. "Recognizing emotions from EEG subbands using wavelet analysis". Em: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. Vol. 2015-Novem. 2015, pp. 6030–6033. ISBN: 9781424492718. DOI: [10.1109/EMBC.2015.7319766](https://doi.org/10.1109/EMBC.2015.7319766).
- [7] G. Chanel, J. Kronegg, D. Grandjean e T. Pun. "Emotion assessment: Arousal evaluation using EEG's and peripheral physiological signals". Em: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4105 LNCS (2006), pp. 530–537. ISSN: 03029743. DOI: [10.1007/11848035](https://doi.org/10.1007/11848035). URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-33750984533&partnerID=tZ0tx3y1>.
- [8] G. Chanel, K. Ansari-Asl e T. Pun. "Valence-arousal evaluation using physiological signals in an emotion recall paradigm". Em: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. 2007, pp. 2662–2667. ISBN: 1424409918. DOI: [10.1109/ICSMC.2007.4413638](https://doi.org/10.1109/ICSMC.2007.4413638).

- [9] R. M. Chapman e H. R. Bragdon. "Evoked Responses To Numerical and Non-Numerical Visual Stimuli While Problem Solving." Em: *Nature* 203 (1964), pp. 1155–1157. ISSN: 0028-0836. DOI: [10.1038/2031155a0](https://doi.org/10.1038/2031155a0).
- [10] *Cortech Solutions Operating ActiveTwo*. <http://www.cortechsolutions.com/getmedia/ded3f9ec-2526-4783-8b0a-da5a330faf1b/Operating-ActiveTwo.aspx>.
- [11] T. F. Cox e M. A. A. Cox. *Multidimensional Scaling, Second Edition*. Vol. 88. 2000, p. 328. ISBN: 1584880945. DOI: [10.1201/9781420036121](https://doi.org/10.1201/9781420036121). URL: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/1584880945>.
- [12] B. J. De Kruif, R. Schaefer e P. Desain. "Classification of imagined beats for use in a brain computer interface". Em: *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*. 2007, pp. 678–681. ISBN: 1424407885. DOI: [10.1109/IEMBS.2007.4352381](https://doi.org/10.1109/IEMBS.2007.4352381).
- [13] J. Demos. *Getting Started with Neurofeedback*. W. W. Norton, 2005. ISBN: 9780393704501. URL: <https://books.google.pt/books?id=p6pVmAEACAAJ>.
- [14] E. Donchin, K. M. Spencer e R. Wijesinghe. "The mental prosthesis: Assessing the speed of a P300-based brain- computer interface". Em: *IEEE Transactions on Rehabilitation Engineering* 8.2 (2000), pp. 174–179. ISSN: 10636528. DOI: [10.1109/86.847808](https://doi.org/10.1109/86.847808).
- [15] G. Dornhege. *Toward brain-computer interfacing*. MIT press, 2007.
- [16] P. Ekman, W. V. Friesen, M. O'Sullivan, A. Chan e et al. "Universals and cultural differences in the judgments of facial expressions of emotion." Em: *Journal of Personality and Social Psychology* 53.4 (1987), pp. 712–717. DOI: [10.1037/0022-3514.53.4.712](https://doi.org/10.1037/0022-3514.53.4.712).
- [17] R. Gifford, L. Steg e J. P. Reser. "Environmental psychology". Em: *IAAP handbooks of applied psychology* (2011), pp. 440–470. DOI: [10.1002/9781444395150.ch18](https://doi.org/10.1002/9781444395150.ch18). URL: [http://ua.lm.worldcat.org/?genre=article&sid=ProQ:&atitle={}&title={}&issn={}&date=2011-01-01&volume={}&issue={}&spage=440&author=Gifford,+Robert;Steg,+Linda;Reser,+Joseph+P.\\$\backslash\\$nhhttp://search.proquest.com/docview/921291559?accountid=17192](http://ua.lm.worldcat.org/?genre=article&sid=ProQ:&atitle={}&title={}&issn={}&date=2011-01-01&volume={}&issue={}&spage=440&author=Gifford,+Robert;Steg,+Linda;Reser,+Joseph+P.\backslashnhhttp://search.proquest.com/docview/921291559?accountid=17192).
- [18] S. W. Golomb. "Shift register sequences - A retrospective account". Em: *Sequences and Their Applications - Seta 2006* 4086 (2006), pp. 1–4. ISSN: 16113349. DOI: [10.1007/11863854__1](https://doi.org/10.1007/11863854__1). URL: GotoISI://000241183000001.
- [19] S. W. Golomb e G. Gong. *Signal Design for Good Correlation*. New York, USA: Cambridge University Press, 2005. ISBN: 978-0-511-15946-6. DOI: [10.1017/CB09780511546907](https://doi.org/10.1017/CB09780511546907).

- [20] H. Gunes e M. Pantic. "Automatic, Dimensional and Continuous Emotion Recognition". Em: *International Journal of Synthetic Emotions* 1.1 (2010), pp. 68–99. ISSN: 1947-9093. DOI: [10.4018/jse.2010101605](https://doi.org/10.4018/jse.2010101605).
- [21] R. J. Hill, C. E. Osgood, G. J. Suci e P. H. Tannenbaum. "The Measurement of Meaning." Em: *American Sociological Review* 23.2 (1958), p. 227. ISSN: 00031224. DOI: [10.2307/2089024](https://doi.org/10.2307/2089024).
- [22] R. D. Hippenstiel. *Detection Theory: Application and Digital Signal Processing*. CRC Press, 2002. ISBN: 0849304342.
- [23] R. Horlings, D. Datcu e L. J. M. Rothkrantz. "Emotion recognition using brain activity". Em: *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing - CompSysTech '08* (2008), p. II.1. DOI: [10.1145/1500879.1500888](https://doi.org/10.1145/1500879.1500888). URL: <http://portal.acm.org/citation.cfm?doid=1500879.1500888>.
- [24] M Islam, T Ahmed, S. S. Mostafa, M. S. U. Yusuf e M Ahmad. "Human emotion recognition using frequency amp; statistical measures of EEG signal". Em: *Informatics, Electronics Vision (ICIEV), 2013 International Conference on* (2013), pp. 1–6. DOI: [10.1109/ICIEV.2013.6572658](https://doi.org/10.1109/ICIEV.2013.6572658).
- [25] R. Jenke, A. Peer e M. Buss. "Effect-size-based electrode and feature selection for emotion recognition from EEG". Em: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2013, pp. 1217–1221. ISBN: 9781479903566. DOI: [10.1109/ICASSP.2013.6637844](https://doi.org/10.1109/ICASSP.2013.6637844).
- [26] R. Jenke, A. Peer e M. Buss. "Feature extraction and selection for emotion recognition from EEG". Em: *IEEE Transactions on Affective Computing* 5.3 (2014), pp. 327–339. ISSN: 19493045. DOI: [10.1109/TAFFC.2014.2339834](https://doi.org/10.1109/TAFFC.2014.2339834).
- [27] V. S. Johnston, D. R. Miller e M. H. Burleson. "Multiple P3s to Emotional Stimuli and Their Theoretical Significance". Em: *Psychophysiology* 23.6 (1986), pp. 684–694. ISSN: 0048-5772. DOI: [10.1111/j.1469-8986.1986.tb00694.x](https://doi.org/10.1111/j.1469-8986.1986.tb00694.x). URL: <http://dx.doi.org/10.1111/j.1469-8986.1986.tb00694.x> \backslash\$nhhttp://download.interscience.wiley.com/cgi-bin/fulltext?ID=119492535{\& }PLACEBO=IE.pdf{\&}mode=pdf.
- [28] A. a. Kosinski. *From Kant to Hilbert: A Source Book in the Foundations of Mathematics*. Vol. 94. 2003, pp. 345–347. ISBN: 019850537X. DOI: [10.1086/379409](https://doi.org/10.1086/379409).
- [29] P. Lang, M. Bradley e B. Cuthbert. *International affective picture system (IAPS): Affective ratings of pictures and instruction manual*. Rel. téc. 2008, Technical Report A–8. DOI: [10.1016/j.epr.2006.03.016](https://doi.org/10.1016/j.epr.2006.03.016). URL: <ftp://dhcp-129-105-171-164.psych.northwestern.edu/OpenShare/ESPN/IAPS1-16/IAPS1-16/IAPSmanual.pdf>.

- [30] Y. P. Lin, C. H. Wang, T. L. Wu, S. K. Jeng e J. H. Chen. "Multilayer perceptron for EEG signal classification during listening to emotional music". Em: *IEEE Region 10 Annual International Conference, Proceedings/TENCON*. 2007. ISBN: 1424412722. DOI: [10.1109/TENCON.2007.4428831](https://doi.org/10.1109/TENCON.2007.4428831).
- [31] S. J. Luck. "Event-Related Potentials". Em: *APA Handbook of Research Methods in Psychology* April 2006 (2012), pp. 1–18. ISSN: 1756-8927. DOI: [10.1016/S0926-6410\(01\)00129-X](https://doi.org/10.1016/S0926-6410(01)00129-X). arXiv: [9780262621960](https://arxiv.org/abs/9780262621960).
- [32] R. K. Maganti e P. Rutecki. *EEG and epilepsy monitoring*. 2013. DOI: [10.1212/01.CON.0000431378.51935.d8](https://doi.org/10.1212/01.CON.0000431378.51935.d8).
- [33] D. McGregor. *The human side of enterprise*. McGraw-Hill, 1960. URL: <https://books.google.pt/books?id=9fA0AAAAAMAAJ>.
- [34] A. Mehrabian e J. A. Russell. *An approach to environmental psychology*. the MIT Press, 1974.
- [35] a Mini, D Palomba, a Angrilli e S Bravi. "Emotional information processing and visual evoked brain potentials." Em: *Perceptual and motor skills* 83.1 (1996), pp. 143–152. ISSN: 0031-5125.
- [36] M. A. Nicolaou, H. Gunes e M. Pantic. "Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space". Em: *IEEE Transactions on Affective Computing* 2.2 (2011), pp. 92–105. ISSN: 19493045. DOI: [10.1109/T-AFFC.2011.9](https://doi.org/10.1109/T-AFFC.2011.9).
- [37] E. Niedermeyer e F. H. L. D. Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Vol. 1. 2004, p. 1309. ISBN: 0781751268. URL: <http://books.google.com/books?id=tndqYGPHQdEC{\&}pgis=1>.
- [38] K. Pearson. "On lines and planes of closest fit to systems of points in space". Em: *Philosophical Magazine Series* 6 2.11 (1901), pp. 559–572. DOI: [10.1080/14786440109462720](https://doi.org/10.1080/14786440109462720).
- [39] J. L. E. E. Rodgers, W. A. Nicewander e L. Toothaker. "Linearly Independent, Orthogonal, and Uncorrelated Variables". Em: 38.2 (1984), pp. 133–134.
- [40] J. Rong, G. Li e Y. P. P. Chen. "Acoustic feature selection for automatic emotion recognition from speech". Em: *Information Processing and Management* 45.3 (2009), pp. 315–328. ISSN: 03064573. DOI: [10.1016/j.ipm.2008.09.003](https://doi.org/10.1016/j.ipm.2008.09.003).
- [41] R Roschmann e W Wittling. "Topographic brain mapping of emotion-related hemisphere asymmetries." Em: *The International journal of neuroscience* 63.1-2 (1992), pp. 5–16. ISSN: 0020-7454. DOI: [10.3109/00207459208986656](https://doi.org/10.3109/00207459208986656). URL: <http://www.ncbi.nlm.nih.gov/pubmed/1342028>.
- [42] D. V. Sarwate e M. B. Pursley. "Crosscorrelation Properties of Pseudorandom and Related Sequences". Em: *Proceedings of the IEEE* 68.5 (1980), pp. 593–619. ISSN: 15582256. DOI: [10.1109/PROC.1980.11697](https://doi.org/10.1109/PROC.1980.11697).

-
- [43] K. Schaaff e T. Schultz. "Towards an EEG-based emotion recognizer for humanoid robots". Em: *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*. 2009, pp. 792–796. ISBN: 9781424450817. DOI: [10.1109/ROMAN.2009.5326306](https://doi.org/10.1109/ROMAN.2009.5326306).
- [44] E. Smith e S. Kosslyn. *Cognitive Psychology: Mind and Brain*. Pearson/Prentice Hall, 2007. ISBN: 9780131825086. URL: <https://books.google.pt/books?id=-x-TQgAACAAJ>.
- [45] B. W. Stahnke. "Primitive Binary Polynomials". Em: 27.124 (1973), pp. 977–980.
- [46] *The Center For The Study Of Emotion And Attention IAPS Message*. <http://csea.phhp.ufl.edu/media/iapsmessage.html>.
- [47] M. Tudor, L. Tudor e K. I. Tudor. "Hans Berger (1873-1941)–the history of electroencephalography". Em: *Acta medica Croatica : casopis Hrvatske akademije medicinskih znanosti* 59.4 (2005), pp. 307–313. ISSN: 1330-0164. URL: <http://www.ncbi.nlm.nih.gov/pubmed/21443395>.
- [48] G. L. Turin. "An Introduction to Matched Filters". Em: *IRE Transactions on Information Theory* 6.3 (1960), pp. 311–329. ISSN: 21682712. DOI: [10.1109/TIT.1960.1057571](https://doi.org/10.1109/TIT.1960.1057571).
- [49] H. Xu e K. N. (Kostas) Plataniotis. "Affect recognition using EEG signal". Em: *IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)* (2012), pp. 299–304. DOI: [10.1109/MMSP.2012.6343458](https://doi.org/10.1109/MMSP.2012.6343458). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6343458>.
- [50] R. E. J. Yohanes, W. Ser e G.-b. Huang. "Discrete wavelet transform coefficients for emotion recognition from EEG signals." Em: *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference 2012* (2012), pp. 2251–4. ISSN: 1557-170X. DOI: [10.1109/EMBC.2012.6346410](https://doi.org/10.1109/EMBC.2012.6346410). URL: <http://www.ncbi.nlm.nih.gov/pubmed/23366371>.
- [51] E. Yourdon e L. L. Constantine. *Structured design: Fundamentals of a discipline of computer program and systems design*. Prentice-Hall, Inc., 1979.



CONVENÇÕES DE ESCRITA

A.1 Nomenclatura Matemática

- Média de uma variável x - \bar{x}
- Período de um segmento - T ;
- Período do sinal adquirido não segmentado - T_{exp} ;
- Amostra proveniente do sistema de aquisição de dados - a
- Segmento - s ;
- Ruído - η ;
- Estimador - \hat{r} ;
- Correlação cruzada - c

A.2 Indices

- Nível - i ;
- Observação - k ;
- Índice genérico - j ;

A.3 Outras Convenções

Quando utilizarmos um valor como K , para descrever o máximo de ocorrência, por exemplo, na equação 3.3, utilizamos por sua vez k para descrever cada ocorrência deste mesmo

valor, como em 3.11. Letras minúsculas para cada ocorrência e maiúsculas para o máximo de ocorrências deste mesmo valor.

Para vetores é utilizada uma letra significativa a negrito. Para matrizes é utilizado uma letra significativa maiúscula a negrito.



DECLARAÇÃO DE CONSENTIMENTO INFORMADO

DEE da FCT-UNL

Laboratório de Sinais



BernEXP02XNUM

Elaboração: 19/08/2016 22:23:00

Impressão: 19/08/2016 22:23:00

Doc/02BET.docx

Figura B.1: Declaração de consentimento informado para uma experiência genérica -
Página traseira

DEE da FCT-UNL

Laboratório de Sinais



Declaração de Consentimento Informado

Esta declaração diz respeito ao procedimento: “Caracterização experimental de micro-comportamento”.

Declaração do Participante:

Declaro que a minha participação é voluntária. Além disso assiste-me o direito de poder revogar este meu compromisso em qualquer altura. Foram-me convenientemente apresentados e por mim aceites, os objectivos, as fases e os diversos pormenores do procedimento experimental em que participo.

Consinto em participar permitindo quer a leitura e registo de dados de potencial eléctrico da superfície da minha pele, quer o registo multimédia. Permito que os meus dados sejam publicados com completo anonimato da minha pessoa.

Identificação:

Número de identificação civil: _____

Sexo: _____ Data de nascimento: ____/ Jun/ 2016

Questões Físicas:

Quanto tempo passou (horas ou dias) desde que:

☐ Tomei café: _____

☐ Ingeri álcool: _____

☐ Tomei um analgésico: _____ Qual? _____

Questões Médicas: Estou a tomar medicação: ☐ Qual: _____
Doença a destacar: _____

Data: ____/ Jun/ 2016

Assinatura:

A preencher pelo experimentador:

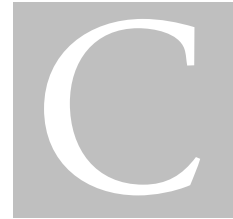
Questões de Meio:

Temperatura: _____ °C.

Humidade relativa: ____ %

Número de matrícula experimental: BemEXP02X006

Figura B.2: Declaração de consentimento informado para uma experiência genérica - Página frontal



PIPELINE DE PROCESSAMENTO - CÓDIGO MATLAB®

C.1 Pipeline

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 % Author: Bernardo Albergaria
5 %
6 % Software: EEG processing - Estimation of image valence
7 %
8 % For more detail consult the following article -
9 %
10 % Latest Version - 18 / 09 / 2016
11 %
12 % How to:
13 %
14 % Inputs: .bdf file, .mat file (translation of the .bdf,
15 % same name), .mat file of the respective random sequence,
16 % kEpochs.mat
17 %
18 % Final Outputs: Filter/Takes/kResults
19 %
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21
22 clc
23 clear all
24 close all
25
26 tic
27

```

```

28 mkdir(' ../Data/values/time' );
29 kFilter = [4 40; 40 100];
30
31 save ../Data/logistic/kFilter kFilter;
32
33 qFilterTakes = numel(kFilter(:,1));
34
35 save ../Data/values/qFilterTakes qFilterTakes;
36
37 qCurrentFilter = 1;
38
39 save ../Data/values/qCurrentFilter qCurrentFilter;
40
41 resultFolder = ' ../Data/finalResults/';
42 mkdir(resultFolder);
43 qCurrentTake = 1;
44
45 save ../Data/values/qCurrentTake qCurrentTake;
46
47 kTakeList = [300 400; 400 500; 300 500];
48
49 save ../Data/logistic/kTakeList kTakeList;
50
51 qTakes = numel(kTakeList(:,1));
52
53 save ../Data/values/qTakes qTakes;
54
55 kElectrodePos = [-92 -72; -69 -72; -92 -36; -57.5 -45; -23 -45; -69 -18; -92 0; -46 0; ...
56 -23 45; -69 18; -92 36; -57.5 45; 46 -90; -69 72; -92 72; 92 -90; ...
57 92 -72; 69 -72; 57.5 -45; 92 -36; 69 -18; 23 -45; 46 0; 92 0; 69 18; ...
58 23 45; 57.5 45; 92 36; 69 72; 92 72; 46 90; 0 0];
59
60 save ../Data/logistic/kElectrodePos kElectrodePos;
61
62 %% Faz00
63
64 clc
65 clear all
66 close all
67 FazA00AzulVerde;
68
69 %% Faz00_5
70
71 clc
72 clear all
73 close all
74 FazA00_5_ConvertTakeList;
75
76 %% Faz01

```

```
77
78 clc
79 clear all
80 close all
81 FazA01LoadData;
82
83 %% Faz02
84
85 clc
86 clear all
87 close all
88 FazA02Limita;
89
90 %% Faz02_5
91
92 clc
93 clear all
94 close all
95 FazA02_5_FiltraData;
96
97 %% Faz03
98
99 clc
100 clear all
101 close all
102 FazA03Combina;
103
104 %% Faz04
105
106 clc
107 clear all
108 close all
109 FazA04Segmenta;
110
111 %% Faz05
112
113 clc
114 clear all
115 close all
116 FazA05Estima;
117
118 %% Faz05.5
119
120 clc
121 clear all
122 close all
123 FazA05_5_SubtraiProprio;
124
125 %% Faz06
126
```

```

127 clc
128 clear all
129 close all
130 FazA06Aplica;
131
132 %% FazPipelineRecursoivo
133
134 load ../Data/values/qCurrentTake;
135 load ../Data/values/qTakes;
136
137 if(qCurrentTake~=qTakes)
138     qCurrentTake = qCurrentTake + 1;
139     save ../Data/values/qCurrentTake qCurrentTake;
140
141     clc
142     clear all
143     close all
144     FazPipelineRecursoivo
145 end
146
147 %% Faz Pipeline Recursivo Filtro
148
149 load ../Data/values/qCurrentFilter
150 load ../Data/values/qFilterTakes
151
152 if(qCurrentFilter~=qFilterTakes)
153
154     qCurrentFilter = qCurrentFilter + 1;
155     save ../Data/values/qCurrentFilter qCurrentFilter;
156     qCurrentTake = 1;
157     save ../Data/values/qCurrentTake qCurrentTake;
158
159     clc
160     clear all
161     close all
162     FazPipelineRecursoivoFilter
163
164 end
165
166 %% Pipeline Close
167
168 qTimeEndPipeline = toc;
169
170 save ../Data/values/time/qTimeEndPipeline qTimeEndPipeline;
171
172 clc
173 clear all
174 close all

```

C.2 Faz 00

Listagem C.1: Faz 00

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 00 - Processing Pipeline
6 %
7 % Inputs: .bdf file
8 %
9 % Outputs: kAzul, kVerde, qFs
10 %
11 % This module loads data from the .bdf file. It saves the data
12 % representing the beginning of the changing of each slide and the
13 % moment when all pixels in the screen have changed into the pixels
14 % of the new slide. It then saves this data and the sample frequency
15 %
16 %
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 %% FazA00AzulVerde - Load da frequencia de amostragem e das transicoes dos slides
20
21 tic
22
23 disp('Faz00 Start - Loading .bdf');
24
25 loadFolder = '../Data/eeg';
26 toDir = strcat(loadFolder,'/*.bdf');
27 sttAa= dir(toDir);
28 qNFiles= numel(sttAa);
29
30 if qNFiles==1
31
32     [strPathstr, strSeqs,strExt] = fileparts(sttAa.name);
33
34 else
35
36     disp('Incorrect bdf file count');
37     keyboard; %Error check
38
39 end
40
41 qdb1FID= fopen(sprintf('%s/%s%s',loadFolder,strSeqs,strExt));
42
43 %Number of bytes in header record
44 qdb1Num= 184;
45 [strTemp, qCount]= fread(qdb1FID,qdb1Num);
46 [strTemp, qCount]= fscanf(qdb1FID,'%8c', 1);

```

```

47 qHeadLen= str2num(strTemp);
48 qC= fseek(qdb1FID,0,-1);          %Rewind
49
50 %Number of Data Records
51 qdb1Num= 236;
52 [strTemp, qCount]= fread(qdb1FID,qdb1Num);
53 [strTemp, qCount]= fscanf(qdb1FID,'%8c', 1);
54 qDataRecordCnt= str2num(strTemp);
55 qC= fseek(qdb1FID,0,-1);          %Rewind
56
57 %Duration of Data Records
58 qdb1Num= 244;
59 [strTemp, qCount]= fread(qdb1FID,qdb1Num);
60 [strTemp, qCount]= fscanf(qdb1FID,'%8c', 1);
61 qDataRecordSecsTot= str2num(strTemp);
62 qC= fseek(qdb1FID,0,-1);          %Rewind
63
64 %Number of Channels
65 qdb1Num= 252;
66 [strTemp, qCount]= fread(qdb1FID,qdb1Num);
67 [strTemp, qCount]= fscanf(qdb1FID,'%4c', 1); %e mesmo 4
68 qChannCnt= str2num(strTemp);
69 qC= fseek(qdb1FID,0,-1);          %Rewind
70
71 %Number of Samples in Each Data Record
72 qdb1Num= 256+ qChannCnt*(16+80+8+8+8+8+80)+(qChannCnt-1)*8;
73 [strTemp, qCount]= fread(qdb1FID,qdb1Num);
74 [strTemp, qCount]= fscanf(qdb1FID,'%8c', 1);
75 qSamp1CntInDataRecord= str2num(strTemp);
76 qC= fseek(qdb1FID,0,-1);          %Rewind
77
78 rStatus= uint8(zeros(qDataRecordCnt*qSamp1CntInDataRecord,3));
79
80 disp('Reading...');
81
82 for nn= 1:qDataRecordCnt
83
84     fprintf('data record = %d of %d\n', nn, qDataRecordCnt );
85
86     for jj=1:qSamp1CntInDataRecord
87
88         qLin= (nn-1)*qSamp1CntInDataRecord+jj;
89         qLastBlockChann= 3*qSamp1CntInDataRecord*(qChannCnt-1);
90         qPrevBlockChann= 3*qSamp1CntInDataRecord*qChannCnt*(nn-1);
91         qOffset= qHeadLen+qPrevBlockChann+qLastBlockChann+3*(jj-1);
92         qC= fseek(qdb1FID,qOffset,-1);
93         [rStatus(qLin,:), qCount]= fscanf(qdb1FID,'%3c', 1);
94
95     end
96

```

```

97 end
98
99 qMask01= uint8(1);
100 qMask02= uint8(2);
101 qMask03= uint8(4);
102
103 disp('Doing...');
104
105 kVerde= uint8(bitand(rStatus(:,1), qMask02));
106 kVerde= uint8(kVerde~=0);
107 kAzul= uint8(bitand(rStatus(:,1), qMask03));
108 kAzul= uint8(kAzul~=0);
109
110 disp('qFs...');
111
112 toLoad=strcat(loadFolder,'/',strSeqs);
113 load (toLoad);
114 qFs= round(1/(double(ALLEEG.times(2)-ALLEEG.times(1))*0.001));
115 mkdir(' ../Data/values');
116
117 save ../Data/logistic/kVerde kVerde;
118 save ../Data/logistic/kAzul kAzul;
119 save ../Data/values/qFs qFs;
120
121 mkdir(' ../Data/values/time');
122
123 qTimePL0 = toc;
124 save ../Data/values/time/qTimePL0 qTimePL0;
125 disp('Faz00 End');

```

C.3 Faz 01

Listagem C.2: Faz 01

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 01 - Processing Pipelin
6 %
7 % Inputs: .mat file from .bdf
8 %
9 % Outputs: mSchs, qMonoChannels, qSignalLen
10 %
11 % This module loads data from the mat file that results from the
12 % conversion of data from .bdf to .mat, using the software:
13 % eeg_lab_4.3
14 % Only the data from all 40 channels is saved
15 %

```

```

16 %
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 %% Faz01LoadData - Load dos Dados
20
21 tic
22
23 disp('Faz01 Start - Loading .mat');
24
25 sttAa= dir(' ../Data/eeg/*.bdf');
26 qNFiles= numel(sttAa);
27
28 if qNFiles==1
29
30     [strPathstr, strSeqs,strExt] = fileparts(sttAa.name);
31
32 else
33
34     disp('Incorrect bdf file count');
35     keyboard; %Error check
36
37 end
38
39 toLoad = strcat(' ../Data/eeg/', strSeqs, '.mat');
40 load (toLoad);
41 rsglData = ALLEEG.data;
42 mSchs = rsglData;
43 qMonoChannels = numel(mSchs(:,1));
44 qSignalLen = numel(mSchs(1,:));
45
46 save ../Data/eeg/mSchs mSchs;
47 save ../Data/values/qMonoChannels qMonoChannels;
48 save ../Data/values/qSignalLen qSignalLen;
49
50 qTimePL1 = toc;
51 save ../Data/values/time/qTimePL1 qTimePL1;
52 disp('Faz01 End');

```

C.4 Faz 02

Listagem C.3: Faz 02

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 02 - Processing Pipeline
6 %
7 % Inputs: kAzul, qFs

```



```

8 %
9 % Outputs: kPosBegin, qSlides
10 %
11 % This module finds where the each slide begins and saves all moments
12 % and the minimum time duration between slides
13 %
14 %
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17 %% FazA02Limites - Limites dos Slides
18
19 tic
20
21 disp('Faz02 Start - Finding Limits');
22
23 load ../Data/logistic/kAzul;
24 load ../Data/values/qFs;
25
26 kAzD= ([0; diff(int8(kAzul))]);
27 kIndex = int32(1:numel(kAzD))';
28 kUpLocs = (kIndex(kAzD>0));
29 kDwLocs = (kIndex(kAzD<0));
30 kAllLocs = kIndex(kAzD~=0);
31 kDiffLocs = diff(kAllLocs);
32 kPosBegin = kAllLocs(kDiffLocs>(qFs*2.43)); %Slides change after 2.43 seconds
33 kPosBegin(floor(numel(kPosBegin)/2)+1)=[];
34 qWindow = min(kDiffLocs((kDiffLocs>10000)));
35 qSlides = numel(kPosBegin);
36 controlFlag = true;
37
38 if controlFlag
39
40     kPosBegin = kPosBegin((qSlides/2 + 1):end); %Usar apenas a metade em que o participante
41         ve os slides
42
43 else
44
45     kPosBegin = kPosBegin(1:(qSlides/2)); %Usar apenas a metade em que o participante nao
46         ve os slides
47
48 end
49
50 qSlides = numel(kPosBegin);
51
52 save ../Data/logistic/kPosBegin kPosBegin;
53 save ../Data/values/qWindow qWindow;
54 save ../Data/values/qSlides qSlides;
55
56 qTimePL3 = toc;
57 save ../Data/values/time/qTimePL3 qTimePL3;

```

```
56 disp('Faz02 End');
```

C.5 Faz 02.5

Listagem C.4: Faz 02.5

```
1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 02_5 - Processing Pipeline
6 %
7 % Inputs: qFS, qMonoChannels, qSignallen, mSchs
8 %
9 % Outputs: mSchs_Non_Filtered, kFilter, kFreqs, kAmps, qFilterLen
10 % kSch(qMonoChannels)
11 %
12 % Modifies: mSchs
13 %
14 % This module filters the data, in order to select a specific range
15 % of frequencies
16 %
17 %
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19
20 tic
21
22 disp('Faz02_5 Start - Filtering eeg');
23
24 %% Faz02_5 - Auxiliary Separation - To decrease processing time
25
26 load ../Data/eeg/mSchs
27 load ../Data/values/qMonoChannels;
28
29 save ../Data/eeg/mSchs_NonFiltered mSchs
30
31 saveFolderAux = '../Data/filter/eeg';
32 mkdir(saveFolderAux);
33
34
35 for cc = 1 : qMonoChannels
36
37     fprintf('Auxiliary Cut %d of %d\n', cc, qMonoChannels);
38     toSave = strcat(saveFolderAux, '/kSch', num2str(cc));
39     kSch = mSchs(cc,:);
40     save(toSave, 'kSch');
41
42 end
43
```

```

44 %% Faz 02_5 - Building Filter
45
46 clc
47 clear all
48 close all
49
50 load ../Data/values/qFs
51 load ../Data/logistic/kFilter;
52 load ../Data/values/qCurrentFilter;
53
54 saveFolder = '../Data/filter';
55 mkdir(saveFolder);
56
57 qFilter = kFilter(qCurrentFilter,:);
58 qHighCut= qFilter(1);
59 qLowCut= qFilter(2);
60 save ../Data/values/qFilter qFilter;
61 qHalfLen= fix(qFs*(1/qHighCut));
62
63 kFreqs= [0;
64 0.01*qHighCut * (2/qFs);
65 0.1* qHighCut * (2/qFs);
66 0.5* qHighCut * (2/qFs);
67 0.9* qHighCut * (2/qFs);
68 0.99*qHighCut * (2/qFs);
69 1*qHighCut * (2/qFs);
70 1.01*qHighCut * (2/qFs);
71 0.9 * qLowCut * (2/qFs);
72 0.99* qLowCut * (2/qFs);
73 1 * qLowCut * (2/qFs);
74 1.01* qLowCut * (2/qFs);
75 (qFs/2) * (2/qFs)
76 ];
77
78 kAmps= [0;
79 0;
80 0;
81 0;
82 0;
83 0;
84 .5;
85 1;
86 1;
87 1;
88 .5;
89 0;
90 0;
91 ];
92
93 fprintf('Building Filter \n');

```

```

94 kFilter= fir2(qHalfLen*2, kFreqs, kAmps);
95 kFilter = kFilter';
96 qFilterLen= numel(kFilter);
97
98 save ../Data/filter/kFilter kFilter
99 save ../Data/filter/kFreqs kFreqs;
100 save ../Data/filter/kAmps kAmps;
101 save ../Data/values/qFilterLen qFilterLen;
102
103 %% Faz 02_5 - Apply Filter
104
105 clc
106 clear all
107 close all
108
109 load ../Data/filter/kFilter;
110 load ../Data/values/qMonoChannels;
111 load ../Data/values/qSignalLen;
112
113 qFilterLen= numel(kFilter);
114 loadFolder = '../Data/filter/eeg';
115
116 for cc=1:qMonoChannels
117
118     fprintf('Filtering Channel %d of %d\n', cc, qMonoChannels);
119     toLoad = strcat(loadFolder, '/kSch', num2str(cc));
120     load(toLoad);
121     kAux= [kSch(1)*ones(qFilterLen,1)', kSch, kSch(end)*ones(qFilterLen,1)'];
122     kSchsConv= conv(kAux, kFilter');
123     qConvLen = numel(kSchsConv);
124     qExtra= qConvLen-qSignalLen;
125
126     if(mod(qExtra,2)~=0)
127
128         fprintf('ERROR - qExtra must be even.\n');
129         keyboard
130
131     else
132
133         qHalfExtra= qExtra/2;
134
135     end
136
137     mSchs(cc,:) = kSchsConv((1+qHalfExtra):(end-qHalfExtra));
138
139 end
140
141 save ../Data/eeg/mSchs mSchs;
142
143 qTimePL2 = toc;

```

```

144 save ../Data/values/time/qTimePL2 qTimePL2;
145 disp('Faz01_5 End');

```

C.6 Faz 03

Listagem C.5: Faz 03

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 03 - Processing Pipeline
6 %
7 % Inputs: mSchs, qMonoChannels
8 %
9 % Outputs: kSchsComb(qChannels), kCombinations, qChannels, qBiChannels
10 %
11 % This module saves all channel data into separate files. It also
12 % builds all bipolar channels after finding every possible combination
13 % between all 40 channels
14 %
15 %
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17
18 %% FazA03Combina - Combina Canais
19
20 tic
21
22 disp('Faz03 Start - Combining');
23
24 load ../Data/eeg/mSchs;
25 load ../Data/values/qMonoChannels;
26
27 saveFolder = '../Data/schsComb';
28 mkdir(saveFolder);
29 kCombinations = nchoosek(1:qMonoChannels,2);
30 qBiChannels = numel(kCombinations(:,1));
31 qChannels = qMonoChannels+qBiChannels;
32
33 for ii = 1:(qChannels)
34
35     fprintf('Combining Channel: %d of %d\n', ii, qChannels);
36
37     if ii<=qMonoChannels
38
39         kSchsComb = mSchs(ii,:);
40         toSave = strcat(saveFolder,'/', 'kSchsComb',num2str(ii),'.mat');
41         save(toSave, 'kSchsComb');
42         continue;

```

```

43
44     end
45
46     kSchsComb = mSchs(kCombinations(ii-qMonoChannels,1),:)-mSchs(kCombinations(ii-
        qMonoChannels,2),:);
47     toSave = strcat(saveFolder,'/', 'kSchsComb', num2str(ii), '.mat');
48     save(toSave, 'kSchsComb');
49
50 end
51
52 save ../Data/logistic/kCombinations kCombinations;
53 save ../Data/values/qChannels qChannels;
54 save ../Data/values/qBiChannels qBiChannels;
55
56 qTimePL4 = toc;
57 save ../Data/values/time/qTimePL4 qTimePL4;
58 disp('Faz03 End');

```

C.7 Faz 04

Listagem C.6: Faz 04

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 04 - Processing Pipeline
6 %
7 % Inputs: kSchsComb(qChannels), kPosBegin, kTakeList, qSlides, qCurrentTake,
8 % qChannels
9 %
10 % Outputs: mSchsSegs(qChannels), qWindow, qBackWindow, qSegmentLen
11 %
12 % This module segments all channels and combinations from a little bit
13 % before the stimuli, until the minimum duration between slides
14 %
15 %
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17
18 %% FazA04Segmenta - Segmentacao dos canais por slides
19
20 tic
21
22 disp('Faz04 Start - Segmenting');
23
24 load ../Data/logistic/kPosBegin;
25 load ../Data/logistic/kTakeList;
26 load ../Data/values/qChannels;
27 load ../Data/values/qSlides;

```

```

28 load ../Data/values/qCurrentTake;
29
30 qBackWindow = kTakeList(qCurrentTake,1);
31 qWindow = kTakeList(qCurrentTake,2);
32
33 save ../Data/values/qWindow qWindow;
34
35 saveFolder = '../Data/schsSegs';
36 loadFolder = '../Data/schsComb';
37 mkdir(saveFolder);
38 qSegmentLen = qWindow+qBackWindow+1;
39
40
41 for ii = 1:(qChannels)
42
43     mSchsSegs = zeros(qSlides,qSegmentLen);
44     fprintf('Segmenting Channel: %d of %d\n', ii, qChannels);
45     toLoad = strcat(loadFolder,'/', 'kSchsComb', num2str(ii), '.mat');
46     load(toLoad);
47
48     for nn = 1 : qSlides
49
50         mSchsSegs(nn,:) = kSchsComb((kPosBegin(nn)-qBackWindow):(kPosBegin(nn)+qWindow));
51         mSchsSegs(nn,:) = mSchsSegs(nn,:) - mean(mSchsSegs(nn,:));
52         mSchsSegs(nn,:) = mSchsSegs(nn,:)/sqrt(mSchsSegs(nn,:)*mSchsSegs(nn,:)');
53
54     end
55
56     toSave = strcat(saveFolder,'/', 'mSchsSegs', num2str(ii), '.mat');
57     save(toSave, 'mSchsSegs');
58     clear kSchsComb;
59
60 end
61
62 save ../Data/values/qSegmentLen qSegmentLen;
63 save ../Data/values/qBackWindow qBackWindow;
64
65 qTimePL5 = toc;
66 save ../Data/values/time/qTimePL5 qTimePL5;
67 disp('Faz04 End');

```

C.8 Faz 05

Listagem C.7: Faz 05

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %

```

```

5 % Module 05 - Processing Pipeline
6 %
7 % Inputs: mSchsSegs(qChannels), kEpochs, kTakeList_Ms, qChannels,
8 % qSlides, qSegmentLen, qWindow, qBackWindow, qCurrentTake
9 %
10 % Outputs: mEstimateFull(qChannels), kOccur, qVals
11 %
12 % This module builds the estimators by averaging all segments related
13 % to the same valence
14 % The module also subtracts the average from the final estimators
15 % After that each estimator is reduced to unitary energy
16 %
17 %
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19
20 %% Faz05Estima - Faz Estimadores Completos - Por Modificar
21
22 tic
23
24 disp('Faz05 Start - Estimating');
25
26 load ../Data/logistic/kEpochs
27 load ../Data/values/qChannels
28 load ../Data/values/qSlides
29 load ../Data/values/qSegmentLen
30 load ../Data/values/qWindow
31 load ../Data/values/qBackWindow
32 load ../Data/values/qCurrentTake
33 load ../Data/logistic/kTakeList_Ms
34
35 loadFolder = '../Data/schsSegs';
36 saveFolder = strcat('../Data/estimateFull','_',num2str(kTakeList(qCurrentTake,1)),'_',
    num2str(kTakeList(qCurrentTake,2)));
37 mkdir(saveFolder);
38 kVals = unique(kImaGrad(:));
39 kFreqOccur = histc(kImaGrad(:),kVals);
40 kOccur = [kVals,kFreqOccur];
41 qVals = length(kVals)-1;
42 mEstimateFull = zeros(qVals-1, qSegmentLen);
43 qAux = 1;
44
45 for cc = 1:qChannels
46
47     for vv = 1:qVals
48
49         fprintf('Estimating Channel: %d of %d; Val %d of %d\n', cc, qChannels, vv, qVals-1);
50
51         if(vv==5)
52
53             continue;

```



```

54
55     end
56
57     toLoad = strcat(loadFolder, '/' , 'mSchsSegs', num2str(cc), '.mat');
58     load(toLoad);
59
60     for ss = 1:qSlides
61
62         if(kImaGrad(ss)>10)
63
64             continue;
65
66         end
67
68         if(kImaGrad(ss)==vv)
69
70             mEstimateFull(qAux,:) = mEstimateFull(qAux,:)+mSchsSegs(ss,:);
71
72         else
73
74             continue;
75
76         end
77
78     end
79
80     qAux = qAux + 1;
81
82 end
83
84 toSave = strcat(saveFolder, '/' , 'mEstimateFull', num2str(cc), '.mat');
85 save(toSave, 'mEstimateFull');
86 mEstimateFull = zeros(qVals-1, qSegmentLen);
87 qAux = 1;
88
89 end
90
91 save ../Data/values/qVals qVals;
92 save ../Data/logistic/k0Occur k0Occur;
93
94 qTimePL6 = toc;
95 save ../Data/values/time/qTimePL6 qTimePL6;
96 disp('Faz05 End');

```

C.9 Faz 05.5

Listagem C.8: Faz 05.5

```

2  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3  %
4  %
5  % Module 06.5 - Processing Pipeline
6  %
7  % Inputs: mSchsSegs(qChannels), mEstimateFull(qChannels), kEpochs,
8  % kOccur, kTake_List_Ms, qChannels, qSlides,qSegmentLen, qVals, qWindow,
9  % qBackWindow, qCurrentTake
10 %
11 % Outputs: mEstimateSpecific(qChannels, qSlides)
12 %
13 % This module subtract each segment from its respective set of
14 % estimators
15 %
16 %
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 %% FazA05_5_SubtraiProprio - Faz Estimadores Especificos
20
21 tic
22
23 disp('Faz05.5 Start - Subtracting Self');
24
25 load ../Data/logistic/kEpochs;
26 load ../Data/logistic/kOccur;
27 load ../Data/values/qChannels;
28 load ../Data/values/qSlides;
29 load ../Data/values/qSegmentLen;
30 load ../Data/values/qVals;
31 load ../Data/values/qWindow;
32 load ../Data/values/qBackWindow;
33 load ../Data/values/qCurrentTake;
34 load ../Data/logistic/kTakeList_Ms;
35
36 saveFolder = strcat('../Data/estimateSpecific','_',num2str(kTakeList(qCurrentTake,1)),'_','_',
    ,num2str(kTakeList(qCurrentTake,2)));
37 loadFolderEstimate = strcat('../Data/estimateFull','_',num2str(kTakeList(qCurrentTake,1))
    ,'_','_',num2str(kTakeList(qCurrentTake,2)));
38 loadFolderSegments = '../Data/schsSegs';
39 mkdir(saveFolder);
40 mEstimateSpecific = zeros(qVals-1, qSegmentLen);
41 mEstimateAll = zeros(qVals-1, qSegmentLen);
42
43 for cc = 1:qChannels
44
45     toLoad = strcat(loadFolderEstimate, '/', 'mEstimateFull', num2str(cc), '.mat');
46     load(toLoad);
47     toLoad = strcat(loadFolderSegments, '/', 'mSchsSegs', num2str(cc), '.mat');
48     load(toLoad);
49     channelFolder = strcat(saveFolder, '/', 'channel', num2str(cc));

```

```

50 mkdir(channelFolder);
51 fprintf('Estimating Channel %d of %d ; General Estimator \n', cc, qChannels);
52
53 for vv = 1:qVals-1
54
55     mEstimateAll(vv, :) = (mEstimateFull(vv,:))/kOccur(vv,2);
56     mEstimateAll(vv, :) = mEstimateAll(vv, :) - mean(mEstimateAll(vv, :));
57     mEstimateAll(vv, :) = mEstimateAll(vv, :)/sqrt(mEstimateAll(vv, :)*mEstimateAll(vv, :));
58
59 end
60
61 for ss = 1:qSlides
62
63     fprintf('Estimating Channel %d of %d ; Specific Slide: %d of %d \n', cc, qChannels,
64         ss, qSlides);
65
66     if(kImaGrad(ss)>10)
67
68         disp('Skipping');
69         continue;
70
71     end
72
73     if(kImaGrad(ss) > 5)
74
75         qAux = kImaGrad(ss) - 1;
76
77     else
78
79         qAux = kImaGrad(ss);
80
81     end
82
83     for vv = 1:qVals-1
84
85         if(vv==qAux)
86
87             mEstimateSpecific(vv, :) = (mEstimateFull(vv,:) - mSchsSegs(ss,:))/(kOccur(vv,2)
88                 -1);
89             mEstimateSpecific(vv, :) = mEstimateSpecific(vv, :) - mean(mEstimateSpecific(vv, :));
90             mEstimateSpecific(vv, :) = mEstimateSpecific(vv, :)/sqrt(mEstimateSpecific(vv, :)*mEstimateSpecific(vv, :));
91
92         else
93
94             mEstimateSpecific(vv, :) = mEstimateAll(vv, :);
95
96         end
97     end
98 end

```

```

95
96     end
97
98     toSave = strcat(channelFolder, '/' , 'mEstimateSpecific', num2str(ss), '.mat');
99     save(toSave, 'mEstimateSpecific');
100
101     end
102
103 end
104
105 qTimePL7 = toc;
106
107 save ../Data/values/time/qTimePL7 qTimePL7
108 disp('Faz05.5 End');

```

C.10 Faz 06

Listagem C.9: Faz 06

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module 06 - Processing Pipeline
6 %
7 % Inputs: mEstimateSpecific(qChannels, qSlides), mSchsSegs(qChannels)
8 % kEpochs, kTakeList_Ms, qChannels, qSlides, qSegmentLen, qVals,
9 % qWindow, qBackWindow, qCurrentTake, qFilter
10 %
11 % Outputs: kResult(qChannels), kError(qChannels), kResultFinal,
12 % kErrorFinal
13 %
14 % This module tests the result of the correlation between each
15 % estimator, of each segment, of each channel, and the corresponding
16 % segment. Each channel contributes with a vote for the valence of
17 % each segment. All errors are calculated
18 %
19 %
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21
22 %% FazA06Aplica - Aplicacao dos Estimadores
23
24 tic
25
26 disp('Faz06 Start - Applying');
27
28 load ../Data/logistic/kEpochs;
29 load ../Data/values/qChannels;
30 load ../Data/values/qSlides;

```

```

31 load ../Data/values/qSegmentLen;
32 load ../Data/values/qVals;
33 load ../Data/values/qWindow;
34 load ../Data/values/qBackWindow;
35 load ../Data/values/qCurrentTake;
36 load ../Data/logistic/kTakeList_Ms;
37 load ../Data/values/qFilter;
38
39 loadFolderEstimate = strcat('../Data/estimateSpecific', '_', num2str(kTakeList(qCurrentTake
    ,1)), '_', num2str(kTakeList(qCurrentTake,2)));
40 loadFolderSegments = '../Data/schsSegs';
41 resultFolder = '../Data/finalResults/';
42 saveFolder = strcat(resultFolder, '_', num2str(qFilter(1)), '_', num2str(qFilter(2)), '/'
    , results_, num2str(kTakeList(qCurrentTake,1)), '_', num2str(kTakeList(qCurrentTake,2)));
43 mVote = zeros(qSlides, qVals-1);
44 mkdir(saveFolder);
45 kAllVotes = zeros(qSlides,1);
46
47 for cc = 1:qChannels
48
49     fprintf('Applying results to channel %d of %d \n', cc, qChannels);
50     toLoad = strcat(loadFolderSegments, '/', 'mSchsSegs', num2str(cc), '.mat');
51     load(toLoad);
52
53     for ss = 1:qSlides
54
55         if(kImaGrad(ss)>10)
56
57             mVote(ss,:) = NaN;
58             continue;
59
60         end
61
62         toLoad = strcat(loadFolderEstimate, '/', 'channel', num2str(cc), '/', '
            mEstimateSpecific', num2str(ss), '.mat');
63         load(toLoad);
64
65         for vv = 1:qVals-1
66
67             mVote(ss,vv) = abs(mSchsSegs(ss,:)*mEstimateSpecific(vv,:));
68
69         end
70
71     end
72
73     kResult = zeros(qSlides,2);
74     kError = zeros(qSlides,1);
75
76     for ss = 1:qSlides
77

```

```

78     if(kImaGrad(ss) > 10)
79
80         continue;
81
82     end
83
84     [~,qMax] = max(mVote(ss,:));
85
86     if(qMax > 4)
87
88         qMaxRect = qMax + 1;
89
90     else
91
92         qMaxRect = qMax;
93
94     end
95
96     kResult(ss,1) = (kImaGrad(ss));
97     kResult(ss,2) = qMaxRect;
98     kError(ss) = kResult(ss,1) - kResult(ss,2);
99
100 end
101
102 kAllVotes(:) = kResult(:,2) + kAllVotes(:);
103 qErrorCount = numel(kError(kError~=0));
104 toSave = strcat(saveFolder, '/', 'kResult', num2str(cc), '.mat');
105 save(toSave,'kResult');
106 toSave = strcat(saveFolder, '/', 'kError', num2str(cc), '.mat');
107 save(toSave,'kError');
108 fprintf('Final error amount for channel %d: %d \n', cc, qErrorCount);
109
110 end
111
112 kAllVotes(:) = round(kAllVotes(:)/qChannels);
113 kResult(:,2) = kAllVotes(:);
114 kError(:) = kResult(:,1) - kResult(:,2);
115 qErrorCount = numel(kError(kError~=0));
116 toSave = strcat(saveFolder, '/', 'kResultFinal', '.mat');
117 save(toSave,'kResult');
118 toSave = strcat(saveFolder, '/', 'kErrorFinal', '.mat');
119 save(toSave,'kError');
120 fprintf('Final error amount for all channel: %d \n', qErrorCount);
121 qTimePL8 = toc;
122 save ../Data/values/time/qTimePL8 qTimePL8;
123 disp('Faz06 End');

```

C.11 Faz xx

Listagem C.10: Faz xx

```

1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 %
5 % Module xx - Processing Pipeline
6 %
7 % Inputs: mSchs, kPosBegin, kSNR, kEpochs, kAzul, qSlides, qSignalLen,
8 % qCurrentSNR, qMonoChannels, qFs
9 % Output: qSNR
10 %
11 % Modifies: mSchs
12 %
13 % This module alter the mSchs data, trading it for random noise +
14 % levelSignal, in order to test the software in different SNR
15 % conditions
16 %
17 %
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19
20 %% FazxxTestSoftwareAuxiliary
21
22 tic
23
24 disp('Fazxx Start - TestSoftwareAuxiliary');
25
26 load ../Data/eeg/mSchs
27 load ../Data/logistic/kPosBegin
28 load ../Data/logistic/kAzul
29 load ../Data/logistic/kSNR
30 load ../Data/values/qSlides
31 load ../Data/values/qSignalLen
32 load ../Data/values/qCurrentSNR
33 load ../Data/logistic/kEpochs
34 load ../Data/values/qMonoChannels
35 load ../Data/values/qFs
36
37 qSNR = kSNR(qCurrentSNR);
38
39 save ../Data/values/qSNR qSNR;
40
41 kAzD= ([0; diff(int8(kAzul))]);
42 kIndex = int32(1:numel(kAzD))';
43 kAllLocs = kIndex(kAzD~=0);
44 kDiffLocs = diff(kAllLocs);
45 qWindow = min(kDiffLocs((kDiffLocs>10000)));
46
47 for cc = 1 : qMonoChannels
48
49     fprintf('Randomizing Channel: %d of %d;\n', cc, qMonoChannels);

```

```

50  mSchs(cc,:) = wgn(qSignalLen, 1, 1);
51  kBigRandom = wgn(qWindow*8,1,1);
52
53  for nn = 1 : 8
54
55      mLevelRandom(nn,:) = kBigRandom(((nn-1)*qWindow+1):(nn)*qWindow);
56
57  end
58
59  for ss = 1 : nSlides
60
61      if(kImaGrad(ss) > 9)
62          continue;
63      end
64
65      if(kImaGrad(ss) > 5)
66          qAux = kImaGrad(ss) - 1;
67      else
68          qAux = kImaGrad(ss);
69      end
70
71      qSNRratio = 10^(qSNR/20);
72      qNoisePower = mSchs(cc, kPosBegin(ss):((kPosBegin(ss)+qWindow)-1)) * mSchs(cc,
73      kPosBegin(ss):((kPosBegin(ss)+qWindow)-1))';
74      qSignalPower = mLevelRandom(qAux,:) * mLevelRandom(qAux,:);
75      qAux = qSNRratio * (qNoisePower/qSignalPower);
76      mSchs(cc, kPosBegin(ss):((kPosBegin(ss)+qWindow)-1)) = mSchs(cc, kPosBegin(ss):((
77      kPosBegin(ss)+qWindow)-1)) ...
78      + sqrt(qAux)*mLevelRandom(qAux,:);
79
80  end
81
82  save ../Data/eeg/mSchs mSchs;
83  disp('Fazxx End');

```


A P Ê N D I C E



PIPELINE - FLUXO DE DADOS

[illegible]

Figura D.1: Fluxo de dados do Pipeline. A bold ficheiros de dados criados. A texto normal ficheiros de dados utilizados. Nomes a azul são modificados e nomes a vermelho não são utilizados.

FILTROS

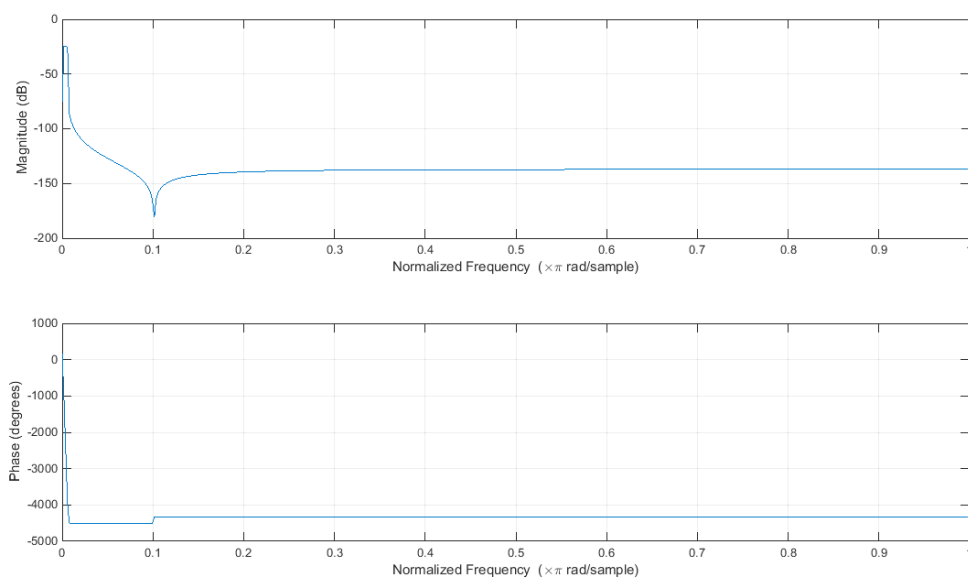


Figura E.1: Característica de magnitude e fase do principal filtro utilizado.

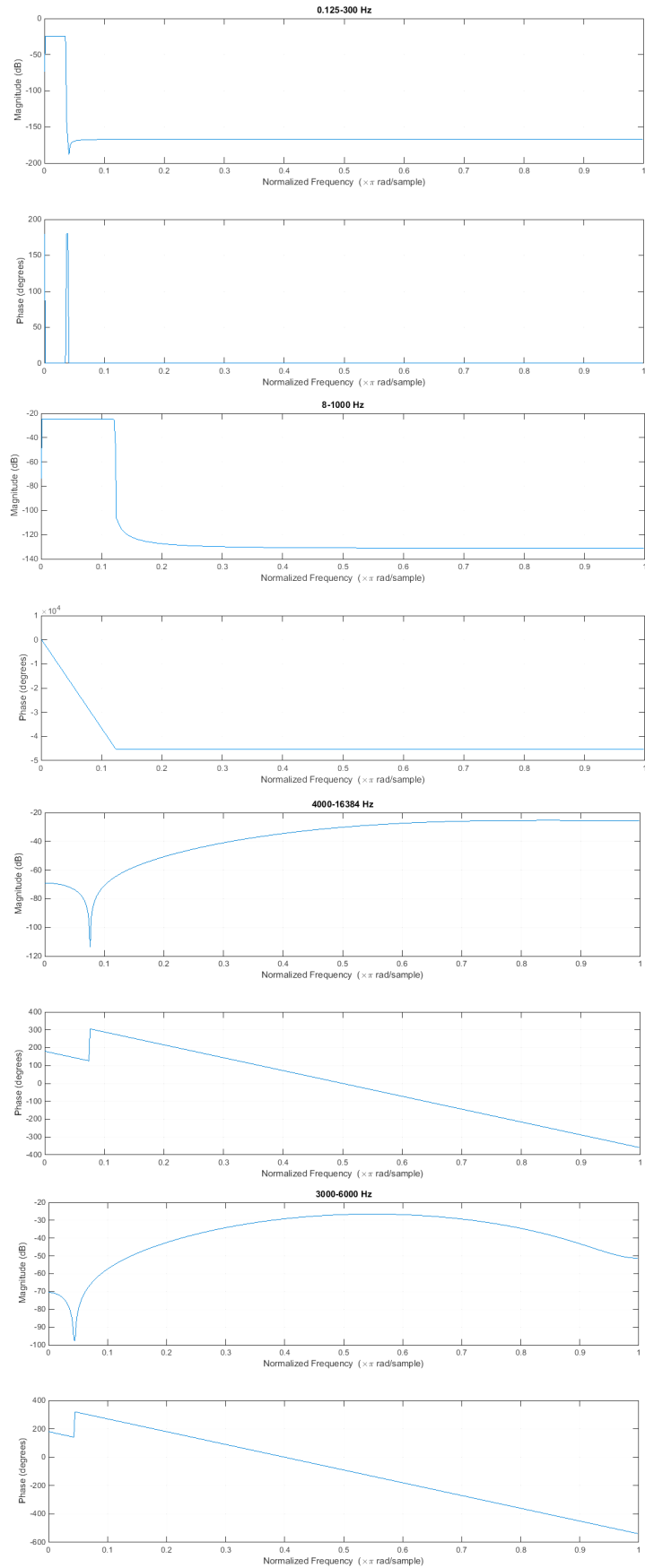


Figura E.2: Característica de magnitude e fase dos outros filtros utilizados.



M-SEQUÊNCIA E SLIDE-SHOW

Seq1	Seq2	Seq3	Nível
0	1	0	3
1	1	0	7
1	1	1	9
0	0	0	4
1	0	0	6
1	1	1	9
0	1	1	1
1	1	0	7
0	0	0	4
0	1	1	1
1	0	0	6
0	1	1	1
0	1	0	3
0	0	1	2
1	0	0	6
0	0	0	4
0	0	0	4
1	1	0	7
1	0	0	6
0	1	0	3
0	1	1	1
1	1	1	9
0	0	1	2

1	0	1	8
0	0	1	2
1	1	0	7
0	1	1	1
0	0	1	2
0	1	1	1
0	1	1	1
0	0	0	4
0	1	0	3
1	0	1	8
1	0	1	8
1	1	1	9
1	0	0	6
1	0	1	8
0	0	0	4
1	1	1	9
1	0	1	8
1	0	0	6
1	1	0	7
0	1	0	3
0	0	0	4
1	0	1	8
1	1	0	7
1	0	1	8
0	1	1	1
1	0	1	8
0	1	0	3
1	0	0	6
1	0	0	6
0	0	1	2
0	0	1	2
0	0	0	4
0	0	1	2
1	1	1	9
0	1	0	3
1	1	1	9
1	1	0	7
1	1	0	7
0	0	1	2
0	1	0	3

0	1	0	3
---	---	---	---

Tabela F.1: Conjunto de 3 sequências geradoras da mensagem de 3 bits e respetiva mensagem criada.



Figura F.1: Slide-Show criado a partir da mensagem da tabela anterior. Cada imagem corresponde a um nível de agrado proveniente da tabela.

As imagens IAPS foram requisitadas através do preenchimento do formulário de informação de investigador presente no site [46]. O processo envolve o envio do formulário e a recepção, caso o requisito seja aceite, de um nome e password de login. Este login permite

aceder e descarregar as imagens IAPS para uso em fins não comerciais por instituições de educação.



RESULTADOS ADICIONAIS - SINGLE BUCKET

Sem Filtro	Moda	Média	Máximo	Canal
take0to50	4	6.96	17	2 e 25
take-300to550	4	7.37	18	6
take-200to1000	0	5.29	16	2
0.125 a 300 Hz	Moda	Média	Máximo	Canal
take0to50	7	7.62	18	5 e 19
take-300to550	5	7.57	19	27
take-200to1000	9	7.53	16	8 e 40
8 a 1000 Hz	Moda	Média	Máximo	Canal
take0to50	8	7.51	19	2 e 25
take-300to550	11	8.45	21	9 e 40
take-200to1000	10	7.61	18	2 e 16
3000 a 6000 Hz	Moda	Média	Máximo	Canal
take0to50	5	6.70	18	1 e 06
take-300to550	4	7.31	20	12 e 25
take-200to1000	2	6.45	18	4 e 28
4000 a 16384 Hz	Moda	Média	Máximo	Canal
take0to50	6	6.91	15	1 e 06
take-300to550	8	7.75	17	21 e 25
take-200to1000	8	7.56	17	10 e 34

Tabela G.1: Resultados adicionais para o Single Bucket

A P Ê N D I C E



TEMPORIZAÇÃO DAS 3 EXPERIÊNCIAS

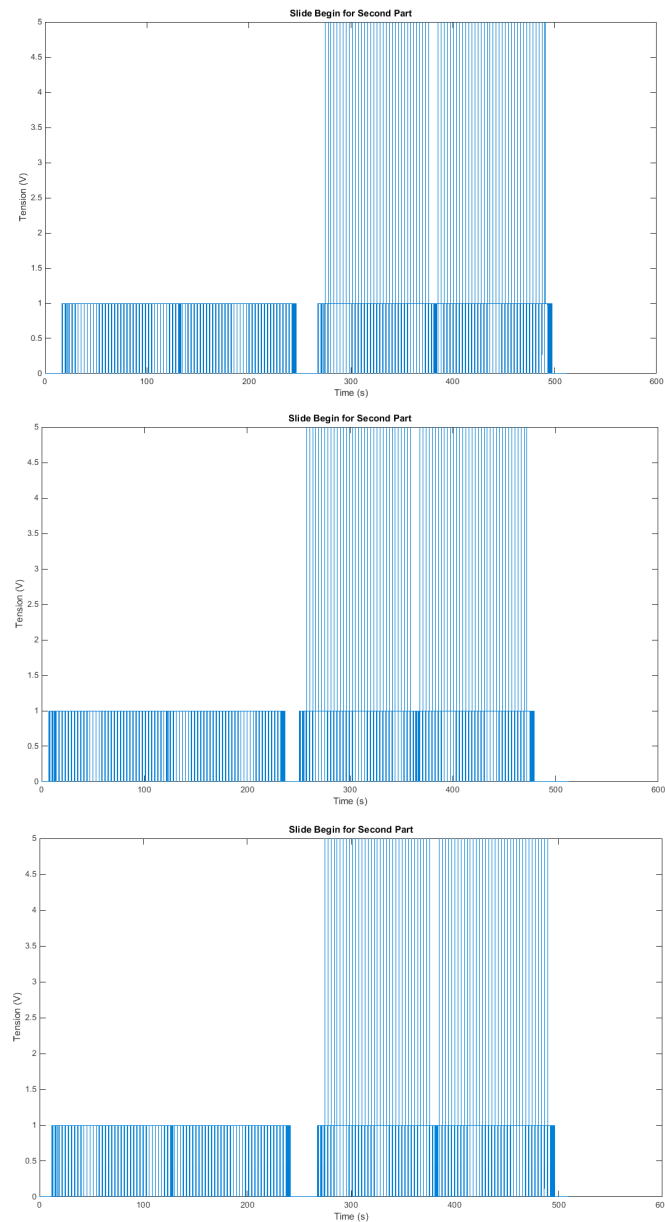


Figura H.1: Momentos detetados para início dos slides para cada experiência - apenas relativo à parte da experiência em que o participante vê os slides. Também é mostrado o sinal kAzul.



RESULTADOS ADICIONAIS - PARTICIPANTES EXPERIMENTAIS

Sem Filtro	Moda	Média	Máximo	Canal
take0to50	15	8.56	17	3 e 24
take-300to550	4	7.80	17	17
take-200to1000	6	7.72	17	5 e 09
0.125 a 300 Hz	Moda	Média	Máximo	Canal
take0to50	7	8.45	18	7 e 20
take-300to550	9	8.06	18	17
take-200to1000	7	7.77	17	10 e 20
8 a 1000 Hz	Moda	Média	Máximo	Canal
take0to50	12	8.43	18	1 e 14
take-300to550	9	8.10	18	25
take-200to1000	9	7.67	17	6 e 11
3000 a 6000 Hz	Moda	Média	Máximo	Canal
take0to50	7	7.81	22	7 e 25
take-300to550	7	7.55	20	14 e 21
take-200to1000	9	7.57	18	7 e 20
4000 a 16384 Hz	Moda	Média	Máximo	Canal
take0to50	6	7.75	20	7 e 25
take-300to550	6	7.98	19	1 e 39
take-200to1000	9	7.77	19	7 e 20

Tabela I.1: Resultados adicionais para a experiência EXP01

APÊNDICE I. RESULTADOS ADICIONAIS - PARTICIPANTES EXPERIMENTAIS

Sem Filtro	Moda	Média	Máximo	Canal
take0to50	8	7.84	18	5 e 34
take-300to550	10	7.18	19	4 e 26
take-200to1000	4	7.01	16	6 e 17
0.125 a 300 Hz	Moda	Média	Máximo	Canal
take0to50	6	7.56	17	13 e 23
take-300to550	8	7.15	18	4 e 26
take-200to1000	7	7.11	17	5 e 10
8 a 1000 Hz	Moda	Média	Máximo	Canal
take0to50	5	7.04	15	20 e 25
take-300to550	2	6.86	17	4 e 11
take-200to1000	12	8.50	20	9 e 11
3000 a 6000 Hz	Moda	Média	Máximo	Canal
take0to50	15	9.23	20	7 e 21
take-300to550	10	7.65	17	33 e 35
take-200to1000	12	8.84	21	4 e 26
4000 a 16384 Hz	Moda	Média	Máximo	Canal
take0to50	14	9.38	21	6 e 15
take-300to550	7	6.95	16	3 e 28
take-200to1000	9	8.14	18	1 e 26

Tabela I.2: Resultados adicionais para a EXP02

Sem Filtro	Moda	Média	Máximo	Canal
take0to50	4	7.82	19	10 e 20
take-300to550	4	7.82	20	9 e 33
take-200to1000	7	7.89	18	4 e 12
0.125 a 300 Hz	Moda	Média	Máximo	Canal
take0to50	5	7.75	18	12 e 20
take-300to550	4	7.83	18	9 e 33
take-200to1000	7	7.93	17	26 e 32
8 a 1000 Hz	Moda	Média	Máximo	Canal
take0to50	5	7.34	17	10 e 20
take-300to550	6	7.68	19	2 e 39
take-200to1000	12	8.47	19	11 e 35
3000 a 6000 Hz	Moda	Média	Máximo	Canal
take0to50	11	8.43	21	14 e 15
take-300to550	6	7.62	18	23 e 28
take-200to1000	5	7.39	19	18 e 35
4000 a 16384 Hz	Moda	Média	Máximo	Canal
take0to50	9	8.41	17	2 e 40
take-300to550	9	8.76	20	23 e 39
take-200to1000	11	9.28	20	30 e 31

Tabela I.3: Resultados adicionais para a EXP03



RESULTADOS DA PRIMEIRA PARTE

	Moda	Média	Máximo	Canal
take-300to400	11	8.40	19	12 e 32
take-400to500	8	8.06	20	26
take-300to500	9	8.06	19	3 e 13

Tabela J.1: Resultados para a primeira parte da experiência EXP01 - participante não vê os slides

	Moda	Média	Máximo	Canal
take-300to400	6	7.49	17	3 e 07
take-400to500	8	8.31	18	13 e 28
take-300to500	7	7.70	19	19 e 38

Tabela J.2: Resultados para a primeira parte da experiência EXP02 - participante não vê os slides

	Moda	Média	Máximo	Canal
take-300to400	2	7.08	16	1 e 34
take-400to500	13	8.99	19	8 e 28
take-300to500	9	7.92	19	8 e 18

Tabela J.3: Resultados para a primeira parte da experiência EXP03 - participante não vê os slides